AD-779 737

DEVELOPMENT OF AN EMPIRICALLY BASED COMPUTER PROGRAM TO PREDICT THE AERODYNAMIC CHARACTERISTICS OF AIRCRAFT VOLUME II. PROGRAM USER GUIDE

Roy T. Schemensky

General Dynamics

Prepared for:

Air Force Flight Dynamics Laboratory

November 1973

Unclassified
Security Classification

AD 779737

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) General Dynamics Corporation Convair Aerospace Division Fort Worth, Texas	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
Development of an Empirically Based Computer Program to Predict the Aerodynamic Characteristics of Aircraft
Volume II: Program User Guide

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report - December 1972 through October 1973

5. AUTHOR(S) (First name, middle initial, last name)
Roy T. Schemensky

6. REPORT DATE November 1973	7a. TOTAL NO. OF PAGES 223	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO. F33615-73-C-3043 b. PROJECT NO. 147601 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) AFFDL-TR-73-144 Volume II 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT
Approved for public release; Distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY AFFDL/FXM

13. ABSTRACT
This report (Volume II, Program User Guide) describes the computer program developed for the evaluation of the aerodynamic characteristics of large aircraft (bombers, tankers, and transports). The program calculates lift, moment, and drag characteristics of aircraft through the subsonic, transonic, and supersonic speed regimes, and includes the capability of calculating the effect of high-lift systems in both free air and in ground effect for landing and takeoff. The input to the computer program requires the configuration geometry and the aerodynamic conditions for which solutions are desired. The program includes the capability of analyzing both fixed-wing and variable-sweep-wing configurations as well as the aerodynamic characteristics of supercritical wing designs. Details of the methods, equations, and substantiating data for this program are contained in Volume I, Empirical Methods. Although this program was developed to handle the bomber, tanker, transport class of aircraft, it is also applicable to fighter type aircraft without maneuver devices.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Aerodynamic Characteristics Drag Prediction Lift Prediction Moment Prediction High-Lift Prediction Computer Program Transport Aircraft Drag Due to Lift Vortex Lift						

ia

Unclassified
Security Classification

☆U.S.Government Printing Office: 1974 — 758-435/606

DEVELOPMENT OF AN EMPIRICALLY BASED COMPUTER PROGRAM TO PREDICT THE AERODYNAMIC CHARACTERISTICS OF AIRCRAFT

Volume II Program User Guide

R. T. SCHEMENSKY

ib

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

ACCESSION for		
NTIS	White Section	☑
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/5 June 1974 — 100

This report was prepared by the Convair Aerospace Division of General Dynamics Corporation, Fort Worth, Texas, for the Air Force Flight Dynamics Laboratory under Contract F33615-73-C-3043, Project 147601. The work reported here was performed in the period December 1972 through October 1973. This report was submitted by the author in October 1973.

The results of this work are documented in two volumes. Volume I presents the methodology developed in this study; Volume II contains the users manual for a computer program which automates these methods.

The work was accomplished under the direction of Mr. J. Kenneth Johnson of the Air Force Flight Dynamics Laboratory (FXM). The author wishes to acknowledge the valuable assistance of Mr. Eugene L. Crosthwait, Convair Aerospace Division, in the development of the empirical methods for the wing lift and wave drag characteristics.

This technical report has been reviewed and is approved.

Philip P. Antonatos

Philip P. Antonatos
Chief, Flight Mechanics Division
Air Force Flight Dynamics Laboratory

TABLE OF CONTENTS

		Page
1.	COMPUTER PROGRAM DESCRIPTION	1
2.	PROGRAM INPUT	5
3.	SAMPLE PROBLEMS	32
4.	PROGRAM AND SUBROUTINE DESCRIPTIONS	57
5.	PROGRAM LISTING	89

1. COMPUTER PROGRAM DESCRIPTION

The Large Aircraft Aerodynamic Prediction Program has been coded in Fortran Extended Version 3.0 to operate on the CDC 6600 computer facility at W-PAFB. This program is also operational on the General Dynamics CDC 6600 computer (procedure code R1T).

The main program controls the calling of three primary overlay programs XINPT, GEOM, and SURVEY. In turn, XINPT calls two secondary overlay programs INPT and NINPT; while SURVEY calls four secondary overlay programs VGEOM, MCRIT, AERO, and LSHL. These programs bring the desired subroutines into the core and direct the calling of the subroutines to make the necessary computations. Figure 1 shows the overlay structure and the subroutines called from these programs.

Card input occurs on file name TAPE5 and data output occurs on file name TAPE6.

The deck structure for a run consists of a job card, system control cards, end-of-record card, binary object decks, end-of-record card, problem data deck, end-of-file card. The structure of the binary object decks and overlay control cards are listed in Table I. The problem data deck is described in the Program Input Section.

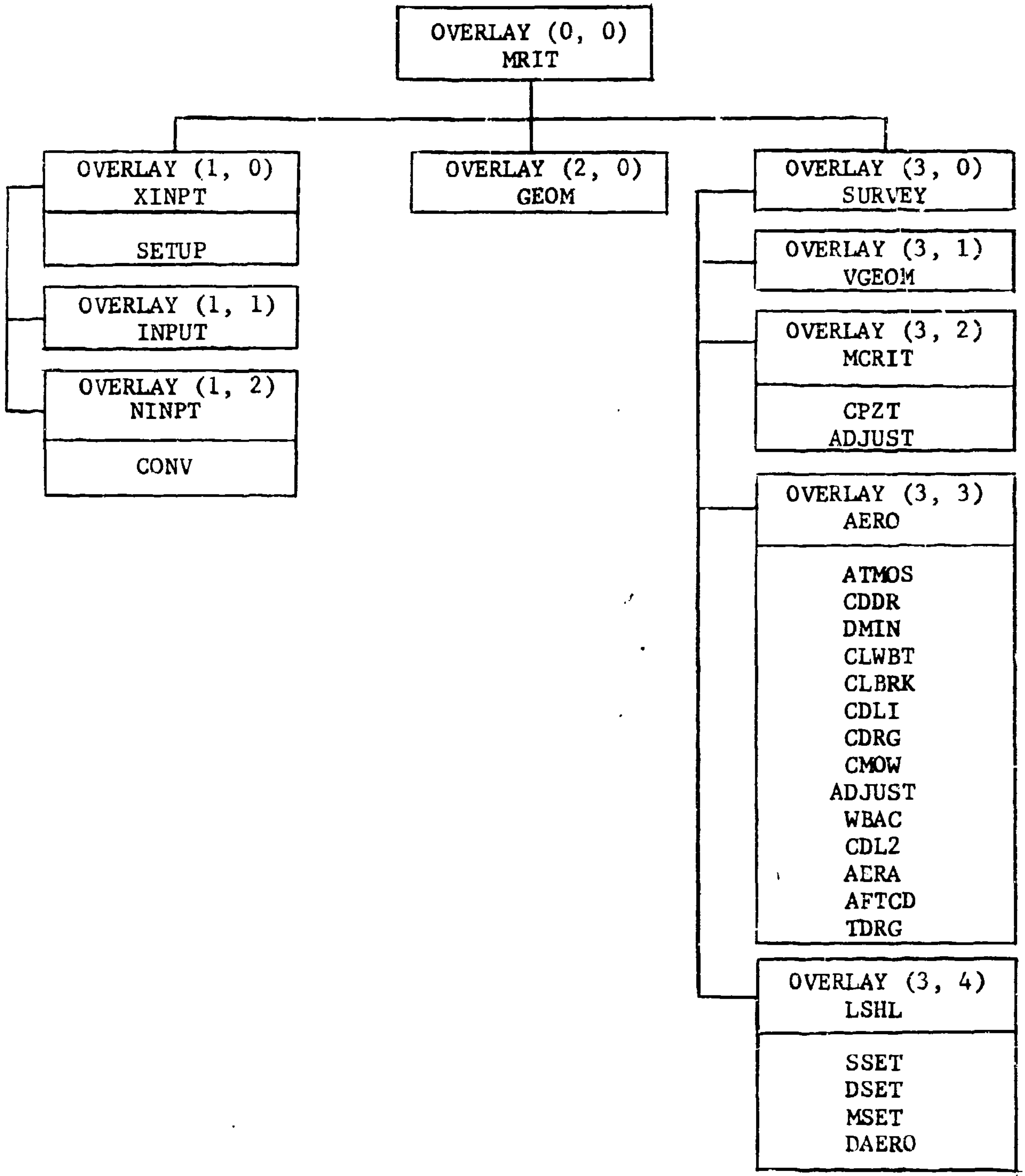

Figure 1 PROGRAM OVERLAY STRUCTURE

Table 1
DECK STRUCTURE FOR OVERLAYS

```
OVERLAY (OULY, 0, 0)
    PROGRAM MRIT
        SUBROUTINE LNTP
        FUNCTION DLNT
        SUBROUTINE TLNT

OVERLAY (1, 0)
    PROGRAM XINPT
        SUBROUTINE SETUP

OVERLAY (1, 1)
    PROGRAM INPT
OVERLAY (1, 2)
    PROGRAM NINPT
        SUBROUTINE CONV

OVERLAY (2, 0)
    PROGRAM GEOM

OVERLAY (3, 0)
    PROGRAM SURVEY
        SUBROUTINE ADJUST
        BLOCK DATA

OVERLAY (3, 1)
    PROGRAM VGEOM

OVERLAY (3, 2)
    PROGRAM MCRIT
        SUBROUTINE CPZT
                   SECT
                   CPUOV

OVERLAY (3, 3)
    PROGRAM AERO
        SUBROUTINE CDDR
                   CLWBT
                   AER2
                   TAIL
                   AALO
                   CLBRK
                   CDL1
                   ADCL
                   KGIN
```

Table 1 (Cont'd)

```
          SUBROUTINE CDL2
                     AERA
                     AFTCD
                     TDRG
                     DMIN
                     FDRG
                     CFEQ
                     FFACT
                     IFACT
                     WDRG
                     CDWN
                     CDWT
                     CDRG
                     BDRG
                     CMOW
                     WBAC
                     ACCR
                     ATMOS
                     PLSQ
                     MTXEQ

OVERLAY (3, 4)
     PROGRAM LSHL
          SUBROUTINE SSET
                     DSET
                     MSET
                     DAERO
```

2. PROGRAM INPUT

The input to this program is described in the following section. Three data types are specified in the card descriptions. Type A indicates that alphabetic or numeric characters should be entered. Type I indicates integers which should be right-adjusted within the specified columns with no decimal point. Type F indicates a real number and may be tabulated anywhere within the specified columns and should include a decimal point. The program input is illustrated by the sample cases in the Sample Problem Section.

Card 1 - Title - Enter any alphanumeric characters to identify each problem. This data is printed out at the top of each aerodynamic survey problem. Columns 1-66.

Card 2 KPRINT(I), Printout Option Indicators - Setting a given KPRINT indicator to 1 causes the program to print out certain types of data. Columns 1-27. KPRINT(11) through (27) are used for diagnostic purposes to dump data generated in various subroutines and are not intended to be used for normal computer runs.

KRPINT and Column No.	Print Out
1	Airfoil ordinates and pressure distribution
2 - 10	Not used
11	Dump Subroutine AER2
12	. GEOM
13	. AALO
14	. CDL2
15	BDRG
16	CLBRK
17	AERA
18	WBAC
19	CFEQ
20	CDWW
21	TAIL
22	CDL1
23	CDDR
24	ADJUST
25	CMOW
26	CPZT
27	Dump Subroutine SSET

Card 3 - Input Control Card - Write "FORMAT" or "NAMELIST" beginning in Column 1 if format or namelist input option is desired.

Cards 4, 5, ..., 27 - Configuration Definition Cards, Format Input Option (For NAMELIST input option go to card 43)

The aircraft geometry is represented as a series of bodies for the fuselage, canopy, and stores; open-nosed bodies for the nacelles; and a series of airfoil surfaces for the wing, tail surfaces, pylon and ventral fins. For cranked or complex wing planforms, the wing is represented as a series of surface panels. Figures 2 through 7 illustrate many of the geometric parameters which are used to define an aircraft configuration. In the

input to the program, those parameters that are marked with an asterisk (*) indicate that if zero or blank is entered the program will calculate that parameter using the available geometric data; otherwise the input value will be used.

Card 4 (Required, all Type I)

Column	Symbol	Definition
4-5	NBODYS	Total number of body types used to represent the configuration
9-10	NNACS	Total number of nacelle types used to represent the configuration
14-15	NSURFS	Total number of airfoil surfaces used to represent the configuration
20	NHT	Enter 1 if a horizontal tail is present
25	NVT	Enter 1 if a single vertical tail is present, 2 for a twin vertical tail
30	ISWP	Variable sweep indicator 0 fixed wing geometry 1 variable sweep wing
34-35	NPNLS	Number of wing panels used to represent the wing surface (≤ 10)
40	IHLS	Enter 1 or 2 if geometry for the high-lift system is to be input. Enter 1 for single high-lift system, enter 2 if the high-lift system is defined by two segments (see Figure 6)
45	IREF	Angle of attack reference indicator 0 referenced to wing root chord plane 1 referenced to fuselage centerline

Card 5 (Required, all Type F)

Column	Symbol	Definition
1-10	SREF	Reference area (ft^2)
11-20	*CMAC	Wing mean aerodynamic chord (in.)
21-30	*XCG	Fuselage station for moment reference point (in.). If zero is input the program will use the quarter-chord of the wing MAC.
31-40	ZCG	Height of moment reference point relative to wing root chord plane
41-50	ROUGHK	Surface roughness height for friction drag (in.)
51-60	FMISC	Miscellaneous drag factor as a percentage of total friction and form drag

Card 6 (Required if NBODYS > 0, all Type F)

1-10	BLEN(I)	Body length (in.)
11-20	BWID(I)	Body width (in.)
21-30	BHGT(I)	Body height (in.)
31-40	*BAWET(I)	Body wetted area (ft^2)
41-50	BQ(I)	Interference factor
51-60	BNO(I)	Number of bodies of this type

Card 7 (Required if NBODYS > 0, all Type F)

1-10	*BAMX(I)	Body maximum cross-sectional area (in^2)
11-20	BABS(I)	Body base area (in^2)
21-30	BLNS(I)	Nose length (in.)
31-40	BLBT(I)	Boattail length (in.)

Card 7 (Continued)

Column	Symbol	Definition
41-50	BASE(I)	Base drag area (in^2)
51-60	BFUS	Aft fuselage upsweep angle (deg.)
61-66	AB	Ratio of aft fuselage width to height in the upswept region

Repeat cards 6 and 7 for I=1 to NBODYS. Note the BFUS and AB are only entered for I=1 which corresponds to the fuselage input.

Card 8 (Required if NNACS > 0, all Type F)

1-10	ELEN(I)	Nacelle length (in.)
11-20	EWID(I)	Nacelle width (in.)
21-30	EHGT(I)	Nacelle height (in.)
31-40	*EAWET(I)	Total wetted area (ft^2)
41-50	EQF(I)	Interference factor
51-60	ENO(I)	Number of nacelles of this type

Card 9 (Required if NNACS > 0, all Type F)

1-10	*EAMX(I)	Nacelle maximum cross-sectional area (in^2)
11-20	EIN(I)	Nacelle inlet area (in^2)
21-30	EXIT(I)	Nacelle exit area (in^2)
31-40	ELNS(I)	Nose length (in.)
41-50	ELBT(I)	Boattail length (in.)

Repeat cards 8 and 9 for I=1 to NNACS.

Card 10 (Required only if NPNLS=1, otherwise go to card 13, all Type F)

Column	Symbol	Definition
1-10	AR	Aspect ratio
11-20	TAPR	Taper ratio
21-30	SWPLE	Leading-edge sweep (deg.)
31-40	SPLAN	Wing planform area (ft^2)
41-50	TWIST	Wing twist (deg.) negative for washout
51-60	WINC	Wing incidence (deg.) relative to fuselage centerline

Card 11 (NPNLS=1, all Type F except as noted)

Column	Symbol	Definition
1-10	TW (Type A)	Type wing section (begin in Col. 1) 64-210 (example 6-series) 00XX-64 (example 4-digit) SUPERCRIT (example supercritical) BICONVEX (example biconvex)
11-20	CAM(1)	Wing camber design C_L (decimal value)
21-30	TOC(1)	Wing thickness (decimal value)
31-40	XLEW	X-position of point on wing leading edge (in.)
41-50	YWW	Y-position of point on wing leading edge (in.)
51-60	YB	Y-distance of intersection of wing with fuselage (in.)

Card 12 (NPNLS=1, all Type F)

Column	Symbol	Definition
1-10	SWMT	Maximum thickness sweep (deg.)
11-20	*CBAR(1)	Exposed wing mean aerodynamic chord (in.)

Card 12 (Continued)

Column	Symbol	Definition
21-30	*AWET(1)	Wetted area (ft^2)
31-40	CONCL	Wing conical camber design C_L

Skip to card 17 after card 12.

Card 13 (NPNLS > 1, all Type F)

Column	Symbol	Definition
1-10	SPLAN	Planform area (ft^2). For fixed wing configurations if zero is input the program will calculate the theoretical planform area by extending the wing panel geometry that is input to the fuselage centerline. For variable sweep configurations the planform area of the movable panel extended to the centerline must be input. For wings with a highly swept inboard strake, the value of the theoretical planform area of the wing, ignoring the strake, should be input.
11-20	TAPR	Taper ratio of wing planform (required input for variable sweep)
21-30	SWPLE	Leading-edge sweep (deg.) of wing planform. (Required input for variable sweep)

Card 14 (NPNLS > 1, all Type F except as noted)

Column	Symbol	Definition
1-10	TW (Type A)	Type wing section
11-20	TWIST	Wing twist (deg.) negative for washout
21-30	WINC	Wing incidence (deg.)
31-40	SWMT	Average maximum thickness sweep for the entire wing planform (deg.)
41-50	CONCL	Wing conical camber design C_L

Card 15 (NPNLS > 1, all Type F)

Column	Symbol	Definition
1-10	CAM(I)	Panel camber design C_L
11-20	TOC(I)	Panel thickness ratio
21-30	XLE(I)	Inboard X-position of panel leading edge (in.)
31-40	YW(I)	Inboard Y-position of panel leading edge (in.)
41-50	CRW(I)	Inboard chord length of panel (in.)
51-60	*CBAR(I)	Panel mean aerodynamic chord length (in.)
61-66	*AWET(I)	Panel wetted area (ft^2)

Repeat card 15 for I=1 to NPNLS

Card 16 (NPNLS > 1, all Type F)

1-10	XLE(NPNLS+1)	Outboard X-position of last panel's leading edge (in.)
11-20	YW(NPNLS+1)	Outboard Y-position of last panel's leading edge (in.)
21-30	CRW(NPNLS+1)	Outboard chord length of panel (in.)

Card 17 (TW = "SUPERCRIT", all Type F)

1-10	XMU	x/c position of maximum thickness on upper surface
11-20	ZMU	upper surface thickness (Zu/c)
21-30	XML	x/c position of maximum thickness on lower surface
31-40	ZML	Lower surface thickness (must be input as a negative value)

Card 17 (Continued)

Column	Symbol	Definition
41-50	ZPTE	Trailing edge slope (deg.)
51-60	ZTHICK	Thickness at trailing edge

Card 18 (ISWP=1, all Type F)

1-10	XPIVOT	X-location of wing pivot (in.)
11-20	YPIVOT	Y-location of wing pivot (in.)
21-30	XAPEX	X-location of centerline apex of movable panel (in.)
31-40	AFTSW	Maximum aft sweep (deg.)
41-50	*AFTCB	Mean aerodynamic chord of movable panel in aft sweep position (in.)
51-60	*AFTOC	Thickness ratio of movable panel in aft sweep position
61-66	*AFTAW	Wetted area of movable panel in aft sweep position

Card 19 (NSURFS ≥ 2, all Type F except as noted) Enter horizontal tail geometry first, then vertical tail followed by any additional airfoil surfaces.

1-10	TS(I) (Type A)	Type airfoil surface
11-20	SCAM(I)	Surface camber design C_L
21-30	STOC(I)	Thickness ratio
31-40	SMTSW(I)	Maximum thickness sweep (deg.)
41-50	SHF(I)	Hinge factor or interference factor

Card 20 (NSURFS ≥ 2, all Type F)

1-10	SWL(I)	Surface leading-edge sweep (deg.)
11-20	SWT(I)	Trailing-edge sweep (deg.)

Card 20 (Continued)

Column	Symbol	Definition
21-30	STAPR(I)	Taper ratio
31-40	SCR(I)	Exposed root chord (in.)
41-50	*SBAR(I)	Exposed mean aerodynamic chord (in.)
51-60	*SAWET(I)	Wetted area (ft^2)

Card 21 (NSURFS ≥ 2, NHT=1, all Type F)

1-10	HTLE	X-location of horizontal tail exposed root chord (in.)
11-20	HTY	Y-location of exposed root chord (in.)
21-30	HTZ	Z-location relative to wing chord plane of horizontal tail (in.)
31-40	HTINC	Incidence of horizontal tail (deg.)

Repeat cards 19 and 20 for I=2 to NSURFS.

Card 22 (NPNLS ≥ 3, all Type F) Since the aerodynamic center calculation is restricted to one or two panel wings, an equivalent two-panel wing must be defined for moment calculation whenever 3 or more panels are used to define the main wing.

1-10	CLE(1)	X-location of leading edge at inboard span station of inboard panel (in.)
11-20	YC(1)	Y-location of leading edge at inboard span station of inboard panel (in.)
21-30	CCR(1)	Chord length of inboard panel at inboard span station (in.)
31-40	CLE(2)	X-location of leading edge at inboard span station of outboard panel (in.)
41-50	YC(2)	Y-location of leading edge at inboard span station of outboard panel (in.)
51-60	CCR(2)	Chord length of outboard panel at inboard span station (in.)

Card 23 (NPNLS ≥ 3, all Type F)

Column	Symbol	Definition
1-10	CLE(3)	X-location of leading edge at outboard span station of outboard panel (in.)
11-20	YC(3)	Y-location of leading edge at outboard span station of outboard panel (in.)
21-30	CCR(3)	Chord length of outboard panel at outboard span station (in.)

Card 24 (Required if IHLS > 0, otherwise go to card 30)

Column	Symbol	Definition
1-20	FLAP (Type A)	Type flap (begin in Col. 1) PLAIN FLAP S.S. FLAP (single slotted) D.S. FLAP (double slotted) T.S. FLAP (triple slotted) INPUT (user defined)
21-40	SLAT (Type A)	Type slat (begin in Col. 21) L.E. FLAP KRUGER SLAT SLAT (slotted type) INPUT
41-45	NF (Type I)	If flap type was "INPUT" indicate number of flap sectional data points to be entered, NF ≤ 5.
46-50	NS (Type I)	If slat type was "INPUT" indicate number of slat sectional data points to be entered, NS ≤ 5.

Card 25 (IHLS > 0, all Type F)

Column	Symbol	Definition
1-10	BF1I	Inboard span station of first flap segment
11-20	BF1O	Outboard span station of first flap segment
21-30	CF1(1)	Flap chord to wing chord ratio of first flap segment

Card 25 (Continued)

Column	Symbol	Definition
31-40	BS1I	Inboard span station of first slat segment
41-50	BS1O	Outboard span station of first slat segment
51-60	CS1	Slat chord to wing chord ratio of first slat segment

Card 26 (IHLS=2, all Type F)

1-10	BF2I	Inboard span station of second flap segment
11-20	BF2O	Outboard span station of second flap segment
21-30	CF2(1)	Flap chord ratio of second flap segment
31-40	BS2I	Inboard span station of second slat segment
41-50	BS2O	Outboard span station of second slat segment
51-60	CS2	Slat chord ratio

Card 27 (FLAP = "D.S." or "T.S.", all Type F) For double-slotted flaps (or triple-slotted) the additional flap chord ratios are then read in.

1-10	CF1(2)	Second flap chord of first flap segment for double- or triple-slotted flap
11-20	CF1(3)	Third flap chord of first flap segment for triple-slotted flap
21-30	CF2(2)	Second flap chord of second flap segment for double- or triple-slotted flap
31-40	CF2(3)	Third flap chord of second flap segment for triple-slotted flap

Card 28 (Required if "INPUT" was used for flap type on card 29, all Type F)

Flap Section Data

Column	Symbol	Definition
1-10	DF(I)	Flap deflection (deg.)
11-20	CFOC(I)	Wing chord to wing chord ratio in extended position
21-30	DCLOF(I)	Change in lift at zero angle of attack due to flap deflection
31-40	DCLMF(I)	Change in maximum lift due to flap deflection
41-50	DCDF(I)	Change in profile drag due to flap deflection
51-60	DCMOF	Change in moment at zero angle of attack due to flap deflection

Repeat card 28 for I=1 to NF.

Card 29 (Required if "INPUT" was used for slat type on card 29, all Type F)

1-10	DS(I)	Slat deflection (deg.)
11-20	CSOC(I)	Wing chord to wing chord ratio in extended position
21-30	DCLOS(I)	Change in lift at zero angle of attack due to slat deflection
31-40	DCLMS(I)	Change in maximum lift due to slat deflection
41-50	DCDS(I)	Change in profile drag due to slat deflection
51-60	DCMOS(I)	Change in moment at zero angle of attack due to slat deflection

Repeat card 29 for I=1 to NS.

Cards 30, 31, 32, and 33 - Problem Control Cards

Card 30 (Required, all Type 1)

Column	Symbol	Definition
4-5	NSURV	Number of high speed lift, drag, moment surveys to be run (NSURV $\leq$ 20)
10	NHLSV	Number of low speed, high lift survey conditions to be run (NHLSV $\leq$ 5)
14-15	NCLAS	Number of evenly spaced C_L values in the high speed survey
21	IT(1)	Transition location indicator, 1st survey
22	IT(2)	Transition location indicator, 2nd survey
⋮	⋮	⋮
40	IT(20)	Transition location indicator, 20th survey

For the ith survey condition, set IT=0 or blank if the flow is to be fully turbulent on all aircraft components, otherwise, set IT equal to the transition pattern number defined by the card 32 input.

Column	Symbol	Definition
41	ITRM(1)	Trim indicator, 1st survey
42	ITRM(2)	Trim indicator, 2nd survey
⋮	⋮	⋮
60	ITRM(20)	Trim indicator, 20th survey

For the ith survey condition, set ITRM=0 or blank if the horizontal tail is fixed, set ITRM=1 if the horizontal tail will vary to trim out the moment.

Card 31 (NSURV > 0, all Type F)

Column	Symbol	Definition
1-10	FMSURV(I)	Mach number for the survey condition
11-20	ALT(I)	Altitude for the survey condition. Reynolds number can be input here instead of altitude by inputing a negative value of RN/ft multiplied by 10^6
21-30	DHSV(I)	Horizontal tail setting (If ITRM(I)=1, the program will calculate the tail setting)
31-40	SWPV(I)	Leading-edge sweep angle (deg.) (Required input only for variable-sweep configurations)
41-50	CLLO(I)	Low C_L value for the survey condition
51-60	CLHI(I)	High C_L value for the survey condition

Repeat card 31 for each survey condition, I=1, NSURV.

Card 32 (IT≠0, 6F10.0 Format)

Each survey has a transition indicator assigned to it. If IT≠0 for any survey, a boundary-layer transition pattern must be defined for each component of the aircraft. Up to five different transition patterns can be used in the surveys. Each transition pattern is numbered sequentially by the order it is input beginning with J=1 for the first transition pattern.

Transition location on a component is indicated by inputing the fraction of the component length where transition occurs.

TRB(J,K1)	Transition location on body K1 (K1 = 1, NBODYS)
TRN(J,K2)	Transition location on nacelle K2 (K2 = 1, NNACS)

Card 32 (Continued)

Column	Symbol	Definition
	TRU(J,K3)	Transition location on upper surface of panel K3 (K3 = 1, NPNLS)
	TRL(J,K3)	Transition location on lower surface of panel K3
	TRS(J,K4)	Transition location on upper and lower surface of airfoil surface K4 (K4 = 2, NSURFS)

For each transition pattern J, first input the body transition location followed in turn by all the nacelle, wing panel, and airfoil surface transition locations. Repeat Card 32 input for each transition pattern J=1, to J_{MAX} where J_{MAX} = maximum value of IT(I=1, 20).

Card 33 (NHLSV > 0, all Type F)

1-10	DFI(1,I)	Flap deflection (deg.) of first flap element
11-20	CPF(I)	Wing chord with flap extended to wing chord with flap retracted ratio
21-30	DSI(I)	Slat deflection (deg.)
31-40	CPS(I)	Wing chord with slat extended to clean wing chord ratio
41-50	DELCD(I)	Landing gear drag
51-60	H(I)	Height of wing $\bar{c}/4$ above ground (ft.)

"PLAIN", "S.S.", or "INPUT" trailing-edge-flap types are single-element flaps. If FLAP = "D.S." or "T.S.", the deflection of each flap element must be input beginning with DFI(1,I) in Columns 1-10, DFI(2,I) in Columns 11-20 and for "T.S." flaps DFI(3,I) in Columns 21-30. The data for CPF(I), CPS(I), etc. are then shifted to the right on the card.

Repeat card 33 for I=1, to NHLSV.

Card 34 - Format Input Control Card - Write "ADJUST" beginning in Column 1 to indicate if adjustment factors are to be applied to some of the aerodynamic parameters predicted by the program. If no adjustment factors are to be read in, write "END OF INPUT" beginning in Column 1 to indicate that all Configuration Definition and Problem Control Cards have been read in and go to card 41.

Cards 35 through 39 - Adjustment Option Cards - The adjustment option allows certain predicted items in the computer procedure to be adjusted to match a desired value. Thus, the predictions can be adjusted to match wind tunnel data, for instance, so that perturbations in geometry for trade studies can be predicted from a firm baseline. An aerodynamic parameter of interest (A_{PRED}) can be adjusted to match an experimental value (A_{EXP}) by the equation

$$A_{EXP} = (A_{PRED}) \cdot YM + YA$$

where YM and YA are input correlation multiplier and adder that are a function of Mach number or lift coefficient.

Card 35 (Type I) Set a given IVAL indicator equal to a non-zero value to identify it as being an aerodynamic parameter to be adjusted. Begin with IVAL=1 for the first parameter to be identified, IVAL=2 for the second parameter, etc.

Column	Symbol	Definition
1	IVAL(1)	Adjust C_{M_O} as a function of Mach number
2	IVAL(2)	Adjust $C_{D_{MISC}}$ as a function of Mach number
3	IVAL(3)	Adjust α_{LO} as a function of Mach number
4	IVAL(4)	Adjust M_{CR} as a function of lift coefficient
24-25	NXVAR	Number of Mach values in the table of Mach function adjust factors (≤ 15)
29-30	NADJ	Number of parameters to be adjusted as a function of Mach number
34-35	NXCL	Number of C_L values in the table of lift function adjust factors (≤ 15)
39-40	NADJ2	Set equal to 1 if M_{CR} is to be adjusted

Card 36 (NXVAR $\geq$ 0, all Type F)

Column	Symbol	Definition
1-10	X(1)	Mach numbers for the table of Mach function adjust factors
11-20	X(2)	
21-30	X(3)	
31-40	X(4)	
41-50	X(5)	
51-60	X(6)	

Repeat card 36 until NXVAR values of X are read in.

Card 37 (NXVAR > 0, all Type F)

1-10	YM(J=1,I=1)	Multiplier factor
11-20	YA(J=1,I=1)	Adder factor
21-30	YM(J=2,I=1)	
31-40	YA(J=2,I=1)	
41-50	YM(J=3,I=1)	
51-60	YA(J=3,I=1)	

Repeat card 37 until NXVAR values are read in (J=NXVAR) for each aerodynamic parameter to be adjusted (I=1 to NADJ).

Card 38 (NXCL > 0, all Type F)

1-10	XCL(1)	C_L values for the table of lift function adjust factors
11-20	XCL(2)	
21-30	XCL(3)	
31-40	XCL(4)	
41-50	XCL(5)	
51-60	XCL(6)	

Repeat card 38 until NXCL values of XCL are read in.

Card 39 (NXCL > 0, all Type F)

Column	Symbol	Definition
1-10	YM(J=1,I=NXVAR+1)	Multiplier factor
11-20	YA(J=1,I=NXVAR+1)	Adder factor
21-30	YM(J=2,I=NXVAR+1)	
31-40	YA(J=2,I=NXVAR+1)	
41-50	YM(J=3,I=NXVAR+1)	
51-60	YA(J=3,I=NXVAR+1)	

Repeat card 39 until NXCL values are read in (J=NXCL).

Card 40 - Format Input Control Card - Write "END OF INPUT" beginning in Column 1 to indicate that all Configuration Definition, Problem Control and Adjust Cards have been read in.

Card 41 - Repeat Problem Control Card - Write "END" to terminate a problem. This clears the input common blocks and reads card 1 as the next card. Write "SAVE" to repeat a problem with only small changes to the geometry or conditions to be run. Additional values will be read in on cards 42 through 50 which will update the previous problem; values not read in will remain unchanged.

Card 42 - New Title - Enter any alphanumeric characters to identify each problem. Columns 1-66.

Card 43 (Required, all Type I)

1	L(1)	BODYS Namelist data to be read in
2	L(2)	NACEL Namelist data to be read in
3	L(3)	WING Namelist data to be read in
4	L(4)	SURFS Namelist data to be read in
5	L(5)	SURV Namelist data to be read in
6	L(6)	STOL Namelist data to be read in
7	L(7)	ADJUST Namelist data to be read in

Set L(1), L(2), ... L(7) equal to 1 if that particular set of Namelist data is to be read in. Only the values that are read in will change the values in storage from the previous problem.

Use of NAMELIST in Input Data

Input data must be in a special form in order to read using a NAMELIST list. The first character (column 1) in each card to be read must be blank. The second character in the first card of a group of data cards must be an $ (dollar sign), immediately followed by the NAMELIST name. The NAMELIST name must be followed by two blanks and must not contain any embeded-ded blanks. This name is followed by data items separated by commas. (A comma after the last item is optional.) The end of a data group is signaled by $. The form of the data items in an input card may be

(1) Variable name = constant

The variable name may be a subscripted array name or a single variable name. The constant may be integer, real, literal, complex, or logical.

(2) Array name = set of constants (separated by commas)

The array name is not subscripted. The set of constants consist of constants of the type integer, real, literal, complex, or logical. The number of constants must be less than or equal to the number of elements in the array. Successive occurrences of the same constant can be represented in the form K * constant.

The variable names and array names specified in the input data set must appear in the NAMELIST list, but the order is not significant.

Each data card must begin with a complete variable or array name or constant. Embedded blanks are not permitted in names or constants. Trailing blanks after integers or exponents are treated as zeros.

NOTE: All data items on a card are entered in Columns 9-66.

Card 44 - BODYS NAMELIST - Namelist input variables

NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS, SREF, CMAC, IREF, ROUGHK, XCG, ZCG, CONCL, BLEN(I), BWID(I), BHGT(I), BAWET(I), BQ(I), BNO(I), BAMX(I), BABS(I), BLNS(I), BLBT(I), BASE(I), BFUS, AB, FMISC

Card 45 - NACEL NAMELIST

NNACS, SREF, ELEN(I), EWID(I), EHGT(I), EAWET(I), EQF(I), ENO(I), EAMX(I), EIN(I), EXIT(I), ELNS(I), ELBT(I)

Card 46 - WING NAMELIST **

NSURFS, ISWP, NPNLS, AR, TAPR, SWPLE, SPLAN, TWIST, WING, TW, CAM(I), TOC(I), XLE(I), YW(I), CRW(I), CBAR(I), AWET(I), SWMT, XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC, AFTAW, SREF, CMAC, CLE(I), YC(I), CCR(I), XLEW, YWW, YB, XCG, CONCL

Card 47 - SURFS NAMELIST **

NSURFS, NHT, NVT, SREF, SBAR(I), TS(I), SCAM(I), STOC(I), SAWET(I), SMTSW(I), SHF(I), SWL(I), SWT(I), STAPR(I), SCR(I), HTLE, HTY, HTZ, HTINC

Card 48 - SURV NAMELIST

NSURV, NHLSV, NCLAS, IT(I), ITRM(I), FMSURV(I), ALT(I), DHSV(I), SWPV(I), CLLO(I), CLHI(I), DFI(I), CPF(I), DSI(I), CPS(I), DELCD(I), H(I), KPRINT(I), TRB(I), TRN(I), TRU(I), TRL(I), TRS(I), DFI2(I), CPF2(I), DSI2(I), CPS2(I)

Card 49 - STOL NAMELIST ***

IHLS, NF, NS, FLAP, SLAT, DF(I), CFDC(I), DCLOF(I), DCLMF(I), DCDF(I), DCMOF(I), DS(I), CSOC(I), DCLOS(I), DCLMS(I), DCOS(I), DCMOS(I), BF1I, BF1O, CF1, BF2I, BF2O, CF2, BS1I, BS1O, CS1, BS2I

Card 50 - ADJUST NAMELIST

IVAL(I), NXVAR, NXCL, X(I), XCL(I), YM(J,I), YA(J,I)

** TW and TS indicate type of airfoil section; the following code is used for NAMELIST input.

CODE	AIRFOIL TYPE	CODE	AIRFOIL TYPE
1	63-0XX	10	00XX-62
2	64-0XX	11	00XX-63
3	65-0XX	12	00XX-64
4	66-0XX	13	00XX-65
5	63A0XX	14	00XX-66
6	64A0XX	15	00XX-33
7	65A0XX	16	00XX-34
8	Supercritical	17	00XX-35
9	Biconvex	18	00XX-93
		19	00XX-94
		20	00XX-95

*** FLAT and SLAT indicate type of high lift system; the following code is used for NAMELIST input.

CODE	FLAP TYPE	CODE	SLAT TYPE
0	No flap	0	No slat
1	Plain flaps	1	Leading-edge flap
2	Single-slotted flap	2	Slotted slat
3	Double-slotted flap	3	Kruger slat
4	Triple-slotted flap	4	Input section data
5	Input section data		

After all the NAMELIST data is read in, card 41 is the next card to be read in.

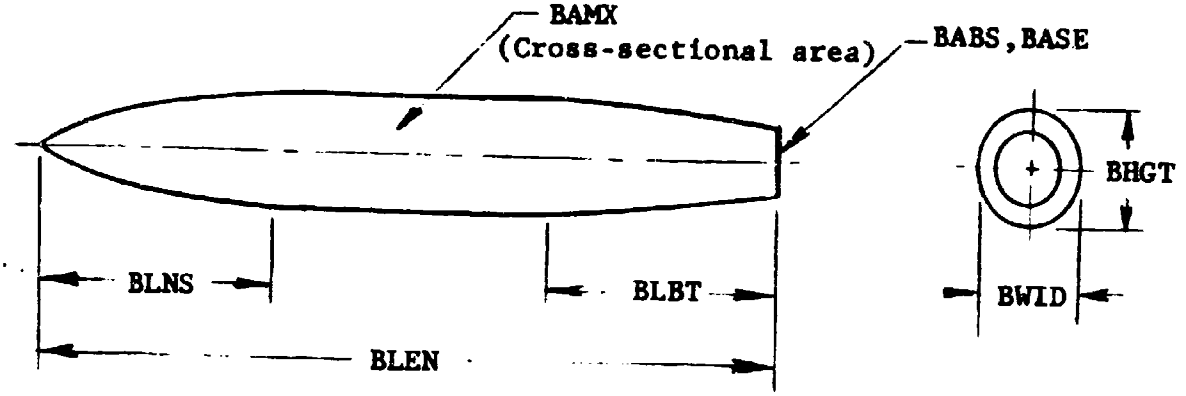

Figure 2 BODIES

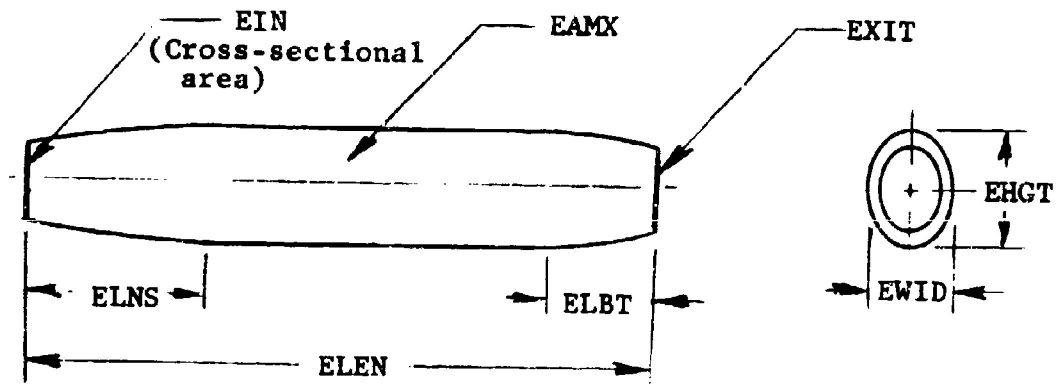

Figure 3 NACELLES

STAPR = SCT/SCR

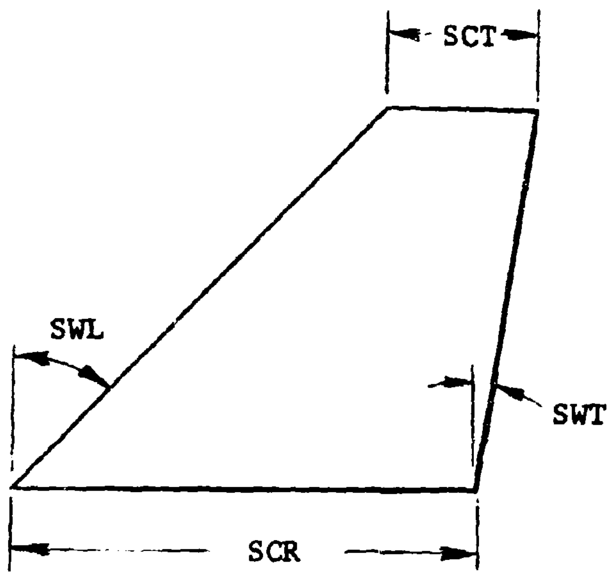

Figure 4 AIRFOIL SURFACE GEOMETRY (OTHER THAN MAIN WING)

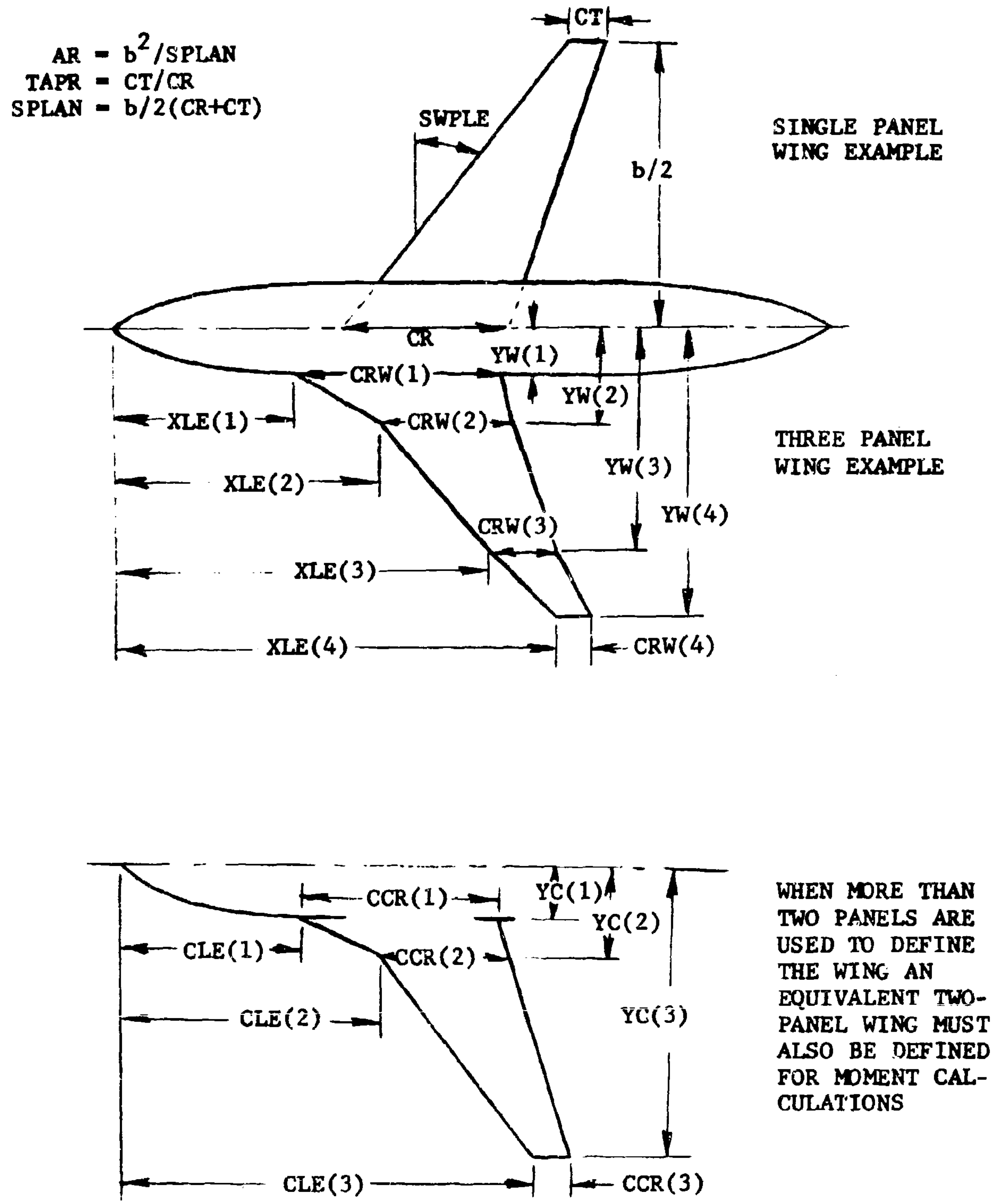

Figure 5 WING SURFACE GEOMETRY

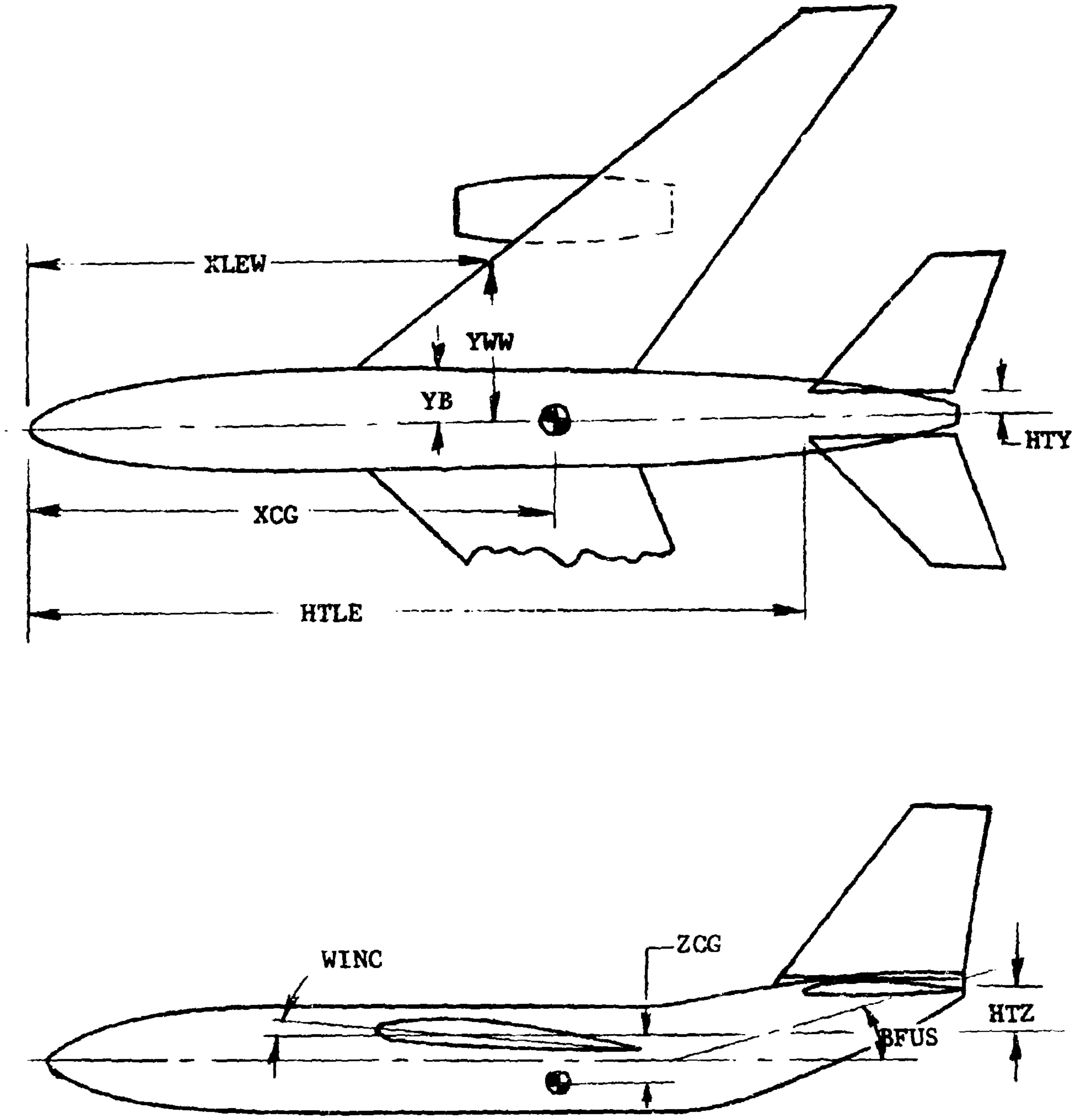

Figure 6 AIRCRAFT GEOMETRY REPRESENTATION

BF1I = $y_1/b/2$
BF1O = $y_2/b/2$
BS1I = $y_5/b/2$
BS1O = $y_6/b/2$
CF1 = C_f/C
CFOC = C'/C

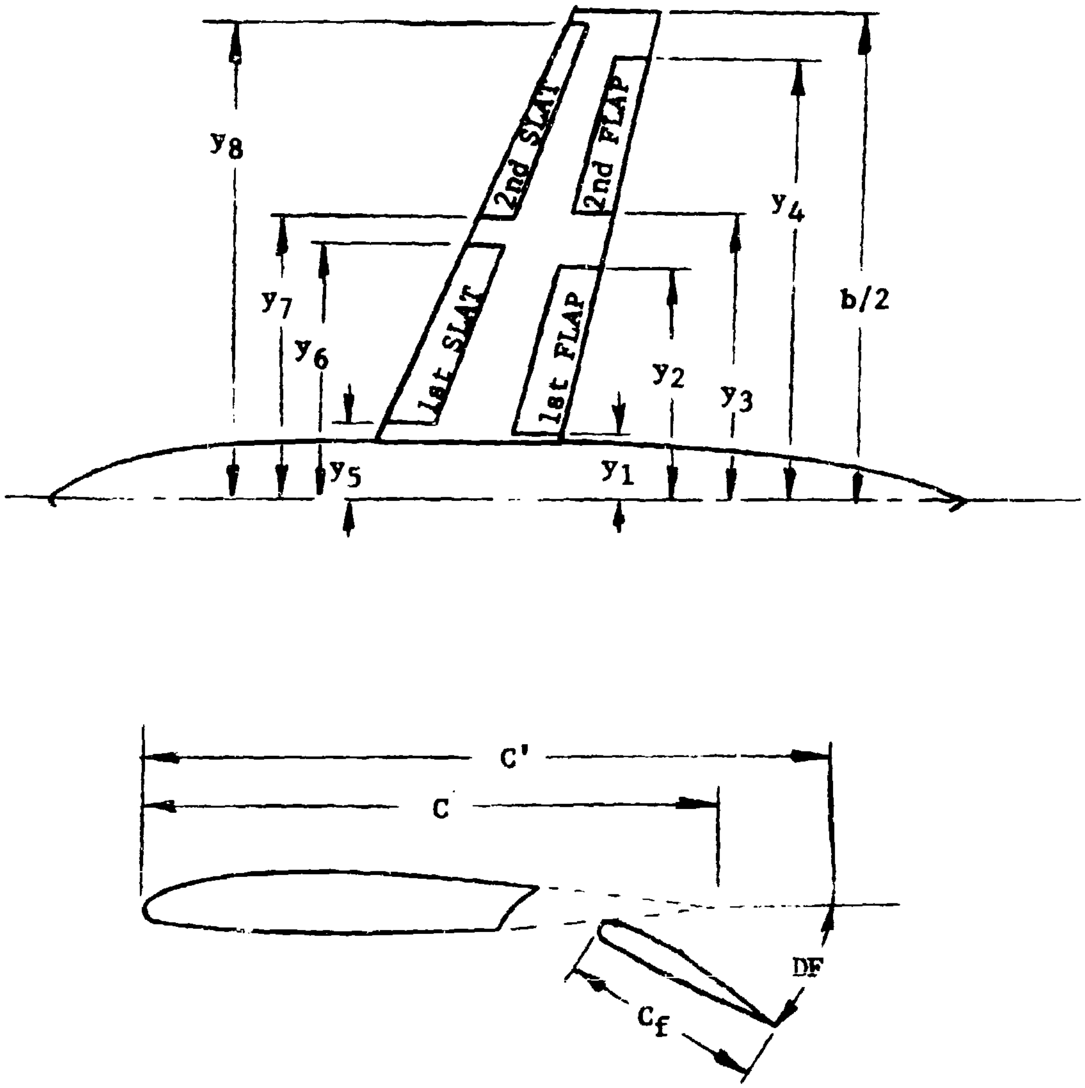

Figure 7 HIGH LIFT SYSTEM GEOMETRY

3. SAMPLE PROBLEMS

Three configurations are analyzed to illustrate the Large Aircraft input techniques and output formats. Problem 1 considers a transport configuration and demonstrates the Format input option and the high-speed lift and drag calculations. Problem 2 demonstrates the input to define the high-lift system for a transport configuration, some dump options and the high-lift system calculations in both free air and in ground effect. Problem 2 also demonstrates the use of the Repeat Problem Control Card to rerun the problem changing the defined configuration from a full-span flap to a partial-span flap high-lift system. Problem 3 demonstrates the Namelist input option for a wing-body configuration.

An abbreviated sample-problem output listing for Problem 1 is presented in pages 35 through 42 corresponding to the configuration defined by the input data in Table 2. The first page of output prints the problem title and the printout option indicators that were set. The next set of output is a listing of some of the input data which occurs when the FORMAT input option is used. The next set of output is a dump of the geometry calculated in Program GEOM which occurs when KPRINT(12) is set equal to 1. A summary of the configuration geometry used in friction drag calculations follows along with a table of the prediction for the critical Mach number versus lift and fuselage aft-end upsweep drag versus wing angle of attack. The last set of data consist of either a tabulation of the predicted drag, moment and angle of attack versus lift for a fixed horizontal tail setting, or a tabulation of drag, tail setting, and angle of attack versus lift for a trimmed (C_M=0) condition.

Several columns of drag versus lift are shown tabulated in the last set of data so that a breakdown of the items contributing to the total drag buildup can be seen. Basic drag due to lift of the wing-body combination is shown in one column, transonic-rise-plus-fuselage aft-end upsweep drag (if any) is shown in another column, while the lift and drag for a zero-horizontal-tail setting are shown in the last two columns on the page. A listing of the items contributing to minimum drag is also shown in the output, along with values of the lift-curve slope, zero-lift angle of attack, polar shape factors, tail-off zero-lift moment, tail-off moment curve slope, separation lift coefficient and maximum lift coefficient.

The input data for sample-problem 2 is shown in Table 3. The sample-problem output listing for problem 2 is presented in pages 44 through 51. Problem 2 set many of the printout option indicators to dump data generated in various subroutines. The dumped data is useful for checking out the program or evaluating the output in detail. The definition of the parameters that are dumped can be determined from the listing of the subroutine. For the high-lift survey problems the final output consists of a table of alpha, C_L, C_D, C_M for both free-air and in-ground effect up to the maximum lift coefficient.

Namelist input data for sample-problem 3 is shown in Table 4 and the output listing is presented in pages 53 through 56.

Table 2 SAMPLE PROBLEM 1

```
C-141A FLIGHT TEST ANALYSIS   NASA CR - 1558                                                   P010001
         1                                                                                     P010002
FORMAT INPUT                                                                                   P010003
    3     1    5     1     1     0     2     0     1                                           P010004
3228.0     266.47     732.88     0.0        0.0        5.00                                    P010005
1590.0     170.0      170.0      4347.52                                                       P010006
22600.0    0.0        204.0      610.0      0.0        7.0        1.0                          P010007
404.0      70.14      70.14      822.0      1.2        2.0                                     P010008
3860.0     0.0        202.0      202.0      0.0                                                P010009
299.84     43.8       43.8       136.8      1.3        1.0                                     P010010
1508.3     0.0        150.0      145.0      0.0                                                P010011
199.22     66.0       66.0       1045.96    1.3        4.0                                     P010012
3421.2     1800.0     1344.96    55.0       100.0                                              P010013
3228.0     0.373                                                                               P010014
0011-63    -5.58      4.89       25.0                                                          P010015
0.4        0.1195     450.0      48.0       380.04     310.3      3134.2                       P010016
0.4        0.11       641.6      404.6      240.7      186.3      2932.2                       P010017
928.03     959.7      131.89                                                                   P010018
64A010.5   0.0        0.105      25.0       1.1                                                P010019
29.0       11.4       0.35       171.15     115.5      893.74                                  P010020
1512.0     0.0        276.0      0.0                                                           P010021
64A013     0.0        0.13       35.0       1.1                                                P010022
38.5       22.5       0.60863    273.17     219.7      819.4                                   P010023
64A010     0.0        0.1        73.0       1.3                                                P010024
73.0       72.0       0.99       200.0      230.0      223.24                                  P010025
64A010     0.0        0.1        73.0       1.3                                                P010026
73.0       72.0       0.99       200.0      200.0      214.88                                  P010027
   12     0    21                           000000111111                                       P010028
0.6        -2.47683   0.0        28.0       0.0        1.0                                     P010029
0.7        -2.47683   0.0        28.0       0.0        1.0                                     P010030
0.75       -2.47683   0.0        28.0       0.0        1.0                                     P010031
0.775      -2.47683   0.0        28.0       0.0        1.0                                     P010032
0.8        -2.47683   0.0        28.0       0.0        1.0                                     P010033
0.81       -2.47683   0.0        28.0       0.0        1.0                                     P010034
0.6        -2.47683              28.0       0.0        1.0                                     P010035
0.7        -2.47683              28.0       0.0        1.0                                     P010036
0.75       -2.47683              28.0       0.0        1.0                                     P010037
0.775      -2.47683              28.0       0.0        1.0                                     P010038
0.         -2.47683              28.0       0.0        1.0                                     P010039
0.81       -2.47683              28.0       0.0        1.0                                     P010040
END OF INPUT                                                                                   P010041
END                                                                                            P010042
```

GENERAL DYNAMICS   CONVAIR AEROSPACE DIVISION   FORT WORTH OPERATION
F60P PROCEDURE P1T   PROBLEM 185829-01   01/10/74   PAGE 0001

EMPIRICALLY BASED COMPUTER PROGRAM

TO PREDICT THE AERODYNAMIC CHARACTERISTICS

OF LARGE AIRCRAFT

C-141A FLIGHT TEST ANALYSIS   NASA CR - 1558

KFRINT(12) = 1

GENERAL DYNAMICS　　CONVAIR AEROSPACE DIVISION　　FORT WORTH OPERATION
6600 PROCEDURE 51T　　PROBLEM 185829-R1　　01/10/74　PAGE 0002

PROBLEM INPUT PARAMETERS

NPCNYS = 7　　NMACS = 1　　NSURFS = 5　　NHT = 1　　NVT = 1　　ISWP = 0　　NPNLS = 2

SREF = 3228.000 SQ.FT.　　CMAC = 266.479 IN.　　FUS. STA. C.G. = 732.889 IN.　　ZCG = .000 IN.

NO.	BLEN	BWID	BHGT	BAWET	BAMX	BABS	BLNS	BLBT	BASE	BQ	BNO
1	1590.000	170.100	170.000	4347.520	22670.000	0.000	204.000	610.000	0.000	1.000	1.000
2	404.000	70.140	70.140	822.000	3861.000	0.000	202.000	292.000	0.000	1.200	2.000
3	299.840	43.800	43.800	136.800	1595.300	0.000	150.000	145.000	0.000	1.300	1.000

NO.	ELEN	EWID	EHGT	EAWET	EAMX	EIN	EXIT	ELNS	ELBT	EQF	ENO
1	199.220	66.000	66.000	1045.960	3421.200	1800.000	1344.960	99.000	100.000	1.300	4.000

AIRFOIL TYPE = 0-11-53　　TWIST = -5.589 DEG.　　INCIDENCE = 4.890 DEG.　　MAX. THICKNESS SWP. = 25.000 DEG.

WING PANEL GEOMETRY

NO.	CAMBER	T/C	XLE	YW	CRW	CBAR	AWET
1	.400	.119	459.000	48.000	399.040	310.300	3134.210
2	.400	.119	641.600	404.600	249.790	186.300	2932.200
			928.030	959.770	131.890		

GEOMETRY FOR ADDITIONAL AIRFOIL SURFACES

AIRFOIL	CAMBER	T/C	SMTSW	SHF	L.E.SWP	T.E.SWP	TAPER	CR	SBAR	SAWET
64A010	0.000	.105	25.000	1.100	29.000	11.499	.350	171.151	115.500	893.740
64A013	0.000	.130	35.000	1.100	38.500	22.513	.609	273.170	219.700	819.400
64A010	0.000	.110	73.000	1.301	73.000	72.019	.990	200.000	200.000	223.240
64A010	0.000	.110	73.000	1.300	73.000	72.019	.990	200.000	200.000	214.880

12 DRAG POLARS TO BE GENERATED AT THE FOLLOWING CONDITIONS

MACH NO.	ALTITUDE	TAIL SETTING	L.E.SWEEP	FROM CL	TO CL
.6000	-2.477	0.000	28.000	0.000	1.000
.7000	-2.477	0.000	28.000	0.000	1.000
.7500	-2.477	0.000	28.000	0.000	1.000
.7750	-2.477	0.000	28.000	0.000	1.000
.8000	-2.477	0.000	28.000	0.000	1.000
.8100	-2.477	0.000	28.000	0.000	1.000
.6000	-2.477	TRIMMED	28.000	0.000	1.000
.7000	-2.477	TRIMMED	28.000	0.000	1.000
.7500	-2.477	TRIMMED	28.000	0.000	1.000
.7750	-2.477	TRIMMED	28.000	0.000	1.000
.8000	-2.477	TRIMMED	28.000	0.000	1.000
.8100	-2.477	TRIMMED	28.000	0.000	1.000

```
GENERAL DYNAMICS          CONVAIR AEROSPACE DIVISION        FORT WORTH OPERATION
5580 PROCEDURE 21T            PROBLEM 185929-01             01/10/74    PAGE 0003

    END OF INPUT              *****
```

GENERAL DYNAMICS CONVAIR AEROSPACE DIVISION FORT WORTH OPERATION
66RP RECCELLFF B1T PROBLEM 105829-01 17/10/74 PAGE 194

GEOMETRY DATA DUMP, COMMON BLKGP1

9.3529	5.7599	6.8457	0.0000	0.0006	0.0000	0.0000	0.0000	0.0000	0.0000
5.4609	2.5528	.1041	.1341	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
505.1379	105.5742	28.6163	28.6163	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3.0185	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	.0500	.1150	.4000	2973.4763	.363	1537.1936
1438.2827	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.4930
.4764	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3295
.3959	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	7.9257
.3367	2228.0000	21.3077	35.5459	54.2342	31.6700	20.0583	10.9908	29.7167	46.2583
1537.1936	1438.2827	2.2979	7.7649	22.9708	7.1983	2075.6246	.493	.4764	.3295
.3959	.4253	.4850	.2264	.2078	-.1319	134.5225	72.8822	76.9661	.3035
.0316									

GENERAL DYNAMICS CONVAIR AEROSPACE DIVISION FORT WORTH OPERATION
6607 PROCEDURE 717 PROBLEM 185829-01 01/10/74 PAGE 0005

CONFIGURATION SUMMARY

	LENGTH (FT.)	WETTED AREA (SQ. FT.)	FR OR T/C	INTERFERENCE FACTOR	MAX. T/C SWEEP (DEG.)
BODY NUMBER 1	132.50000	4347.52000	9.35294	1.00000	
BODY NUMBER 2	73.66667	822.99100	5.75991	1.20000	
BODY NUMBER 3	24.98667	136.80100	6.84566	1.30000	
NACELLE NO. 4	16.60167	1045.95000	3.01848	1.30000	
WING PANEL NO. 1	25.85833	3134.20000	.11950		25.00001
WING PANEL NO. 2	15.52501	2932.20000	.11900		25.00001
AIRFOIL SURFACE NO. 2	9.62510	893.74000	.10509	1.10000	25.00001
AIRFOIL SURFACE NO. 3	18.30833	819.40000	.13000	1.10000	35.00001
AIRFOIL SURFACE NO. 4	16.66667	223.24000	.10000	1.30000	73.00003
AIRFOIL SURFACE NO. 5	16.66667	214.80000	.10000	1.30000	73.00003

4-DIGIT AIRFOIL SECTION

MACH CRITICAL TABLE

CL	MACH CRITICAL
0.0000	.8310
.1000	.8186
.2000	.8062
.3000	.7918
.4000	.7763
.5000	.7600
.6000	.7421
.7000	.7226
.8000	.7013
.9000	.6787
1.0000	.6559

FUSLAGE AFT-END UPSWEEP DRAG

WING ANGLE	DFLCD
0.00000	.00188
1.00000	.00154
2.00000	.00125
3.00000	.00101
4.00000	.00080
5.00000	.00063
6.00000	.00048
7.00000	.00035
8.00000	.00024
9.00000	.00016
10.00000	.00010
11.00000	.00007
12.00000	.00006
13.00000	.00006
14.00000	.00007
15.00000	.00007
16.00000	.00007
17.00000	-.00001
18.00000	-.00007
19.00000	-.00017
20.00000	-.00029

C-141A FLIGHT TEST ANALYSIS   NASA CR - 1558
MACH NO. = .700   RN/FT = 2.4760E+06   L.E. SWEEP ANGLE = [illegible] DEG.   TAIL DEFL. (DH) = [illegible] DEG.

CL	TOTAL CD	CM	ALPHA	CD LIFT	CD R+AFT	CL AT DH=0	CD AT DH=0
0.000	.01716	.06785	-5.05	.00025	.01194	.0400	.01776
.050	.01658	.06072	-4.61	.00002	.01178	.05[illegible]	.01668
.100	.01656	.05749	-4.18	.00015	.01164	.10[illegible]	.01656
.150	.01671	.04620	-3.74	.00034	.01158	.15009	.01671
.200	.01713	.03[illegible]5	-3.31	.00089	.01137	.21009	.01713
.250	.01752	.03145	-2.87	.00170	.01125	.25[illegible]	.01782
.300	.01878	.02798	-2.43	.00277	.01114	.30009	.01878
.350	.02001	.01647	-2.00	.00410	.01014	.35[illegible]	.02001
.400	.02150	.00890	-1.56	.00569	.00994	.40[illegible]	.02159
.450	.02326	.00129	-1.12	.00757	.00985	.45001	.02326
.500	.02528	-.00637	-.69	.00964	.00977	.50[illegible]	.02528
.550	.02756	-.01408	-.25	.01270	.00969	.55009	.02756
.600	.03012	-.02183	.18	.01403	.00961	.60900	.03012
.650	.03293	-.02962	.62	.01751	.00955	.65[illegible]	.03293
.700	.03611	-.03744	1.06	.02005	.00948	.70[illegible]	.03611
.750	.03936	-.04530	1.49	.02416	.00942	.75[illegible]	.03936
.800	.04297	-.05720	1.93	.02772	.00977	.80[illegible]	.04297
.850	.04723	-.06112	2.37	.03164	.00068	.85[illegible]	.04729
.900	.05261	-.06908	2.80	.03582	.00192	.90009	.05261
.950	.05925	-.07705	3.24	.04026	.00412	.95[illegible]	.05925
1.000	.06710	-.08506	3.68	.04496	.00727	1.0000	.06710

CLA = .11460 PER DEG.   ALC =-5.05155 DEG.   CLDB = 1.29936
K = .00186   CFLCL = .06851   CLMAX = 1.28090
CMO = -.03642   CM/CL = .1154

DRAG BREAKDOWN -----		FUSLAGE	BODIES	NACELLES	WING	HORIZ TAIL	VERT TAIL	SURFACES
FRICTION	= .00950	.00273	.00062	.00075	.00422	.01969	.0[illegible]58	[illegible]31
FORM	= .00178	.00023	.00019	.00009	.00098	.00012	.0[illegible]13	.00[illegible]15
INTERF	= .00208	.00004	.00017	.00025	.00105	.0[illegible]025	.00019	.0[illegible]13
WAVE	=0.00000	0.00000	0.00000	0.00000	0.00000	[illegible]	[illegible]	[illegible]
BASE	=0.00000	0.00000	0.00000					
CAMBER	= .00000							
DRAG RISE	=0.00000							
MISC	= .00067							

CDMIN = .0148[illegible]

C-141A FLIGHT TEST ANALYSIS   NASA CR - 1558
MACH NO. = .700   RN/FT = 2.4768E+06   L.E. SWEEP ANGLE = 0.00 DEG.   TRIMMED CONDITION

CL	TOTAL CD	DH	ALPHA	CD LIFT	CD R+AFT	CL AT DH=0	CJ AT DH=0
.021	.01646	1.47893	-5.05	.00025	.00194	0.00000	.01716
.068	.01632	1.31974	-4.61	.00032	.00178	.05000	.01660
.116	.01623	1.15997	-4.18	.00065	.00164	.10000	.01656
.164	.01645	.99966	-3.74	.00034	.00150	.15000	.01671
.212	.01699	.83982	-3.31	.00089	.00137	.20000	.01713
.259	.01775	.67748	-2.87	.00170	.00125	.25000	.01782
.307	.01876	.51565	-2.43	.00277	.00114	.30000	.01878
.355	.02011	.35737	-2.00	.00410	.00104	.35000	.02001
.403	.02151	.19065	-1.55	.00569	.00094	.40000	.02150
.450	.02326	.02752	-1.12	.00753	.00085	.45000	.02326
.498	.02525	-.13601	-.69	.00964	.00077	.50000	.02528
.546	.02749	-.29990	-.25	.01200	.00069	.55000	.02756
.594	.02997	-.46414	.18	.01463	.00061	.60000	.03012
.641	.03277	-.62870	.62	.01751	.00055	.65000	.03293
.689	.03557	-.79757	1.06	.02065	.00048	.70000	.03601
.737	.03893	-.95873	1.49	.02406	.00042	.75000	.03936
.784	.04234	-1.12415	1.93	.02772	.00037	.80000	.04297
.832	.04649	-1.28981	2.37	.03164	.00068	.85000	.04729
.880	.05163	-1.45570	2.80	.03582	.00192	.90000	.05261
.927	.05878	-1.62178	3.24	.04026	.00412	.95000	.05925
.975	.06568	-1.78806	3.68	.04496	.00727	1.00000	.06710

CLA = .11460 PER DEG.   ALC =-5.05055 DEG.   CLOB = 1.29036
K = .05106   DELCL = -.6891   CLMAX = 1.28[illegible]
CMO = -.03012   CM/CL = .1154

DRAG BREAKDOWN -----		FUSLAGE	BODIES	NACELLES	WING	HORIZ TAIL	VERT TAIL	SURFACES
FRICTION	= .01050	.00233	.00062	.00075	.00422	.00069	.00158	.00[illegible]31
FCSP	= .00178	.0[illegible]	.00019	.00009	.00098	.00012	.00013	.00005
INTERF	= .00208	.0000[illegible]	.00017	.00025	.00105	.00025	.00019	.00013
WAVE	=0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
BASE	=0.00000	0.00000	0.00000					
CAMBER	= .00000							
DRAG RISE	=0.00000							
MISC	= .00067							

CDMIN = .01487

Table 3 SAMPLE PROBLEM 2

```
NACA TN D-7034   4-ENGINE STOL TRANSPORT  (HIGH WING)                005573P060001
1           1111 1111     1111                                       005573P060002
FORMAT INPUT                                                         005573P060003
    1     0     3     1     1     0     1     1                      005573P060004
5.143       11.28       27.09       -3.73                            005573P060005
68.268      10.404      9.504                                        005573P060006
            0.0         12.15       25.2        0.0                  005573P060007
7.0         0.3         29.0        5.143       -5.05       2.85     005573P060008
66-012      0.0         0.12        24.27       14.7692     4.75     005573P060009
25.0                                                                 005573P060010
66-012      0.0         0.12        7.5         1.0                  005573P060011
12.5        -7.5        0.44        9.816                            005573P060012
58.255      0.0         17.67       0.0                              005573P060013
66-012      0.0         0.12        22.0        1.0                  005573P060014
27.0        6.0         0.52        14.628                           005573P060015
D.S. FLAP               SLAT                                         005573P060016
0.134       1.0         0.16        0.134       1.0         0.15     005573P060017
0.38                                                                 005573P060018
    0     1                                                          005573P060019
30.0        30.0        1.16        19.0        1.1                  005573P060020
1.14                                                                 005573P060021
END OF INPUT                                                         005573P060022
SAVE                                                                 005573P060023
NACA TN D-7034  PARTIAL SPAN FLAP                                    005573P060024
     1                                                               005573P060025
 $STOL  BF10=0.75$                                                   005573P060026
END OF PROBLEM                                                       005573P060027

                                                               CC = 00027
```

EMPIRICALLY BASED COMPUTER PROGRAM

TO PREDICT THE AERODYNAMIC CHARACTERISTICS

OF LARGE AIRCRAFT

NACA TN D-7034 4-ENGINE STOL TRANSPORT (HIGH WING)

```
KPRINT( 1) =  1
KPRINT(11) =  1
KPRINT(12) =  1
KPRINT(13) =  1
KPRINT(14) =  1
KPRINT(16) =  1
KPRINT(17) =  1
KPRINT(18) =  1
KPRINT(19) =  1
KPRINT(24) =  1
KPRINT(25) =  1
KPRINT(26) =  1
KPRINT(27) =  1
```

GENERAL DYNAMICS   CONVAIR AEROSPACE DIVISION   FORT WORTH OPERATION
6600 PROCEDURE 11T   PROBLEM CP5573-06   13730773   PAGE 0003

PROBLEM INPUT PARAMETERS

NBODYS = 1   NNACS = 0   NSURFS = 3   NHT = 1   NVT = 1   ISWP = 0   NPNLS = 1

SREF = 5.143 SQ.FT.   CHAC = 11.280 IN.   FUS. STA. C.G. = 27.090 IN.   ZCG = -3.730 IN.

NO.	BLEN	BWID	BHGT	BAWET	BAMX	BABS	BLNS	BLBT	BASE	BQ	BNO
1	60.268	10.400	9.504	-0.000	-0.000	0.000	12.150	25.200	0.000	1.000	1.000

ASPECT RATIO = 7.000   TAPER RATIO = .3000   L.E.SWEEP = 29.000   PLANFORM AREA = 5.14 SQ.FT.
TWIST = -5.050 DEG.   INCIDENCE = 2.850 DEG.
AIRFOIL TYPE = 66-.12   CAMBER = 0.0000   THICKNESS = .1200
WING LOCATION (X,Y) = 24.2700   14.7692FUSLAGE INTERSECTION = 4.7500
MAX. THICKNESS SWP. = 25.000   CBAR = -0.0000   AWET = -0.000

GEOMETRY FOR ADDITIONAL AIRFOIL SURFACES

AIRFOIL	CAMBER	T/C	SMTSW	SHF	L.E.SWP	T.E.SWP	TAPER	CR	SBAR	SAWET
66-012	0.000	.120	7.500	1.000	12.500	-7.000	.440	9.816	-0.000	-0.000
66-012	0.000	.120	22.000	1.000	27.000	6.000	.520	14.628	-0.000	-0.000

END OF INPUT   *****

GENERAL DYNAMICS 6600 PROCEDURE R1T — CONVAIR AEROSPACE DIVISION, PROBLEM 065573-06 — FORT WORTH OPERATION, 10/30/73, PAGE 0004

GEOMETRY DATA DUMP, COMMON BLKG01

6.8654	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4.4023	3.1234	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.5271	1.3414	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.1000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
.5061	.4454	.3809	.2418	.1319	.1206	0.0000	0.0000	0.0000	0.0000
1.0000	0.0000	0.3300	1.0000	0.0100	0.0000	0.0000	4.1472	.3819	4.1472
1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.5061
1.0000	0.0000	0.0000	1.0300	0.0000	0.0000	0.0000	0.0000	0.0000	.3809
.3000	5.1433	.9400	1.2338	2.0225	1.1969	0.0000	.3956	2.6642	7.0000
4.1472	1.0000	0.0000	6.5411	0.0000	0.0000	0.0000	.5061	0.0000	0.0000
0.0000	.4454	.5061	.2418	.2138	-.1383	5.1343	2.6591	3.3647	.5549
1.0000									

GENERAL DYNAMICS CONVAIR AEROSPACE DIVISION FORT WORTH OPERATION
6600 PROCEDURE R1T PROBLEM 605573-06 10/30/73 PAGE 0005

CONFIGURATION SUMMARY

	LENGTH (FT.)	WETTED AREA (SQ. FT.)	FR OR T/C	INTERFERENCE FACTOR	MAX. T/C SWEEP (DEG.)	
BODY NUMBER 1	5.68900	11.96929	6.86537	1.00000		
WING PANEL NO. 1	.06346	8.47136	.12000		25.00001	
AIRFOIL SURFACE NO. 2	.61865	3.11925	.12000	1.00000	7.50000	
AIRFOIL SURFACE NO. 3	.95724	2.73793	.12000	1.00000	22.00001	
CFEQ INPUT	7.1007E+06	1.0000000	5.6590000	0.0000000	-0.0000000	
CFEQ OUTPUT	.0022403	.0022403	0.0000000	.9859024	7.1007E+06	0.
CFEQ INPUT	7.1007E+06	1.0000000	.8634567	0.0000000	-0.0000000	
CFEQ OUTPUT	.0030117	.0030117	0.0000000	.9859024	7.1007E+06	0.
CFEQ INPUT	7.1007E+06	1.0000000	.8634587	0.0000000	-0.0000000	
CFEQ OUTPUT	.0030117	.0030117	0.0000000	.9859024	7.1007E+06	0.
CFEQ INPUT	7.1007E+06	1.0000000	.6180525	0.0000000	-0.0000000	
CFEQ OUTPUT	.0031854	.0031854	0.0000000	.9859024	7.1007E+06	0.
CFEQ INPUT	7.1007E+06	1.0000000	.9572406	0.0000000	-0.0000000	
CFEQ OUTPUT	.0029607	.0029607	0.0000000	.9859024	7.1007E+06	0.
CFEQ INPUT	6.3906E+06	.9000000	5.6890000	0.0000000	-0.0000000	
CFEQ OUTPUT	.0023031	.0023031	0.0000000	.9884409	6.3906E+06	0.
CFEQ INPUT	6.3906E+06	.9000000	.8634567	0.0000000	-0.0000000	
CFEQ OUTPUT	.0030996	.0030995	0.0000000	.9884409	6.3906E+06	0.
CFEQ INPUT	6.3906E+06	.9000000	.8634587	0.0000000	-0.0000000	
CFEQ OUTPUT	.0030996	.0030996	0.0000000	.9884409	6.3906E+06	0.
CFEQ INPUT	6.3906E+06	.9000000	.6186535	0.0000000	-0.0000000	
CFEQ OUTPUT	.0032791	.0032791	0.0000000	.9884409	6.3906E+06	0.
CFEQ INPUT	6.3906E+06	.9000000	.9572406	0.0000000	-0.0000000	
CFEQ OUTPUT	.0030468	.0030468	0.0000000	.9884409	6.3906E+06	0.
AER2 DUMP	.90000	.06000	.14618	.40759	1.03332	5.14300
	5.14300	.91300	.78781	.90781	1.12000	.92833
	.91255	.10000	.12000	.30000	7.00000	0.00000
	0.00000	0.00000	.13194	1.00000	.38091	
AER2 DUMP	.10000	.08101	.07639	1.00000	1.03332	5.14300
	5.14300	.91300	.78781	.90781	1.12000	.92833
	.91255	.10000	.12000	.30000	7.00000	0.00000
	0.00000	0.00000	.13194	1.00000	.38091	
AER2 DUMP	.90000	.01403	.11337	.41667	1.00000	1.52706
	5.14300	.91944	.77110	.89110	1.12000	.99899
	.91944	.10000	.12000	.44000	4.49234	0.00000
	0.00000	0.00000	0.00000	1.00000	.34499	
AALO DUMP ALO = 2.029	ALOC =-0.000		ALOT = 1.085	ALOI = .945		
.9.000	.44539	.12000	0.00000	-5.05000	47.00153	.13194
.30000	.06030	.03625	.01110	.01949	.17823	1.10369
2.85000	0.00000	2.85000				
CFEQ INPUT	7.8108E+06	1.1000000	5.6890000	0.0000000	-0.0000000	
CFEQ OUTPUT	.0021799	.0021799	0.0000000	.9831659	7.8108E+06	0.
CFEQ INPUT	7.8108E+06	1.1000000	.8634587	0.0000000	-0.0000000	
CFEQ OUTPUT	.0029278	.0029278	0.0000000	.9831659	7.8108E+06	0.
CFEQ INPUT	7.8108E+06	1.1000000	.8634587	0.0000000	-0.0000000	

```
          CFEQ OUTPUT      .0029278     .0029278      0.0000000      .9831659    7.8108E+06      0.
          CFEQ INPUT     7.8108E+06    1.1000000      .6186535      0.0000000   -0.0000000
          CFEQ OUTPUT      .0030962     .0030962      0.0000000      .9831659    7.8108E+06      0.
          CFEQ INPUT     7.8108E+06    1.1000000      .9572406      0.0000000   -0.0000000
          CFEQ OUTPUT      .0028784     .0028784      0.0000000      .9831659    7.8108E+06      0.
          AER2 DUMP       1.11100       .12569        .12237         .99652      1.03332       5.14300
                          5.14300       .91300        .78781         .90781      1.12000        .92833
                           .91255       .10000        .12000         .30000      7.00000       0.00000
                          1.00000      0.00000        .13194        1.00000       .38091
          AER2 DUMP        .10000       .08131        .07639        1.00000      1.03332       5.14300
                          5.14300       .91300        .78781         .90781      1.12000        .92833
                           .91255       .10000        .12000         .30000      7.00000       0.00000
                          0.00000      0.00000        .13194        1.00000       .38091
          AER2 DUMP       1.10000       .23298        .11139         .99727      1.00000       1.52786
                          5.14300       .91944        .77110         .89110      1.12000        .99899
                           .91944       .10000        .12000         .44000      4.40234       0.00000
                          0.00000      0.00000      0.00000        1.00000       .04499
   AALO DUMP      ALO =   .830      ALOC =-0.000        ALOT = 0.000         ALOI =   .890
            1.10000        .44539        .12000      0.00000        -5.15000     47.60153        .13194
             .30000        .12569        .00625       .01854          .03572       .17823       1.16369
            2.85000       0.00000       2.85000
          CFEQ INPUT     1.4201E+06     .2000000     5.6890000      0.0000000   -0.0000000
          CFEQ OUTPUT      .0033466     .0033466      0.0000000      .9993988    1.4201E+06      0.
          CFEQ INPUT     1.4201E+06     .2000000      .8634587      0.0000000   -0.0000000
          CFEQ OUTPUT      .0042091     .0042091      0.0000000      .9993988    1.4201E+06      0.
          CFEQ INPUT     1.4201E+06     .2000000      .8634587      0.0000000   -0.0000000
          CFEQ OUTPUT      .0042091     .0042091      0.0000000      .9993988    1.4201E+06      0.
          CFEQ INPUT     1.4201E+06     .2000000      .6186535      0.0000000   -0.0000000
          CFEQ OUTPUT      .0044769     .0044769      0.0000000      .9993988    1.4201E+06      0.
          CFEQ INPUT     1.4201E+06     .2000000      .9572406      0.0000000   -0.0000000
          CFEQ OUTPUT      .0041309     .0041309      0.0000000      .9993988    1.4201E+06      0.
          AER2 DUMP        .20000       .08141        .07678        1.00000      1.03332       5.14300
                          5.14300       .91300        .78781         .90781      1.12000        .92833
                           .91255       .10000        .12000         .30000      7.00000       0.00000
                          0.00000      0.00000        .13194        1.00000       .38091
          AER2 DUMP        .10000       .08101        .07639        1.00000      1.03332       5.14300
                          5.14300       .91300        .78781         .90781      1.12000        .92833
                           .91255       .10000        .12000         .30000      7.00000       0.00000
                          0.00000      0.00000        .13194        1.00000       .38091
          AER2 DUMP        .20000       .02106        .07094        1.00000      1.00000       1.52796
                          5.14300       .91944        .77110         .89110      1.12000        .99899
                           .91944       .10000        .12000         .44000      4.4023       0.00000
                          0.00000      0.00000      0.00000        1.00000       .04499
   AALO DUMP      ALO = 2.054       ALOC =-0.000        ALOT = 1.147         ALOI =   .906
             .20000        .44539        .12000      0.00000        -5.05000     25.97673        .13194
             .30000        .06141        .00625       .01511          .02643       .17823       1.10369
            2.85000       0.00000       2.85000
          CLBKK DUMP
             .91263        .98166       9.84114       0.88314          .00000      2.23260      0.80000
            1.01050       1.29200       2.20200        .50000          .00000     16.55988      3.04894
            2.23671        .45000       1.27913       0.00000          .64831      0.00000       .22678
```

```
GENERAL DYNAMICS          CONVAIR AEROSPACE DIVISION        FORT WORTH OPERATION
6600 PROCEDURE R1T           PROBLEM 005573-06              10/30/73   PAGE 0007

     1.32500       1.31000      1.28500      1.28000      1.11345     -.23390      6.12546
  CMOW DUMP
      .02304       0.00000       .62265      1.01693      0.00000
   -5.05000        .44539       .33053      -.00449                  6.54112       .12000
    4.14724                                               .90003     4.00000      1.00000
  WBAC DUMP
        .55349      -.85180       .03625       .09785      .37009
        .61132       .03522       .20009      1.88800      .17838      .43528
     CULZ DUMP
      .20000       0.00000      0.00000       .06079      0.00000
      .93166        .04654      1.88057                                .19579       .91263
  AERA DUMP    HIGH ASPECT RATIO
     0.00000       2.35362      7.00000      4.28671      1.11346       .74785      3.61475
      .13277       2.35362
  SSET DUMP
       3.25725      1.52946       .29814      -.64410      -.04650       .69475      1.00000
        .19657      1.16000      1.10000      2.01537      1.24190      -.00000       .77000
        .54970      -.00000      4.99879      4.31481       .00000       .87952       .04993
      -.00000        .27533       .33018       .00000      1.58553      -.43737      -.20674
      -.00000       -.09290      -.07126       .00000      2.33593      -.14022      1.37270
      1.69561        .91000      3.00000      2.00000     30.00000     30.00000      0.00000
      19.00000       .13793       .32759      0.00000       .13636      7.22400       .07890
```

HIGH LIFT CONSTANTS

```
ZKB   =   .014413        ZKC   =  1.306458        ZKS   =   .798797
CLUCI=   .824555        DC1F  =  3.257277        DC1MF=  1.529456
DCLLS=  -.046497        DC1MS =   .694746        DCLUS=  -.030626
DCLLF=  2.201473

SWPHL=   .391467        ZMD   =   .809729        ZKA   =  1.276285
ZKF   =   .164200        CDC   =  1.206210        SWPLE=   .506145
AR    =  7.000000        TPR   =   .390000        AK    =   .160792
DCL   =  0.000000

DCOMIN =   .329376       CDI   =   .046185        DCOS  =  0.000000
DCDF  =   .423149        DCL   =  0.000000        DCLF  =   .576828
CDC   =   .060142

DCMLF =  -.874933
DCMCS =   .198419
DCMC  =  -.831826
```

GENERAL DYNAMICS
6600 PROCEDURE R1T

CONVAIR AEROSPACE DIVISION
PROBLEM 005573-06

FORT WORTH OPERATION
10/30/73 PAGE 0069

HIGH LIFT SURVEY

ALPHA	CL	CD	CM	AGRD	CLG	CDG	CMG
0.00	1.95979	.44972	-.72670	.201	1.96283	.39114	-.74092
1.00	2.17493	.46795	-.77305	1.291	2.18319	.40266	-.79993
2.00	2.17970	.49790	-.82034	2.380	2.19321	.42533	-.86027
3.00	2.26852	.53279	-.86847	3.457	2.28714	.45323	-.92137
4.00	2.34138	.57490	-.91740	4.518	2.36492	.48772	-.98297
5.00	2.39828	.62389	-.96709	5.559	2.42651	.53160	-1.04481
6.00	2.43923	.68031	-1.01750	6.579	2.47188	.58694	-1.10671
7.00	2.46422	.76934	-1.06857	7.576	2.50161	.66284	-1.16847
8.00	2.47326	.88962	-1.12027	8.548	2.51389	.77427	-1.22997
9.00	2.46634	1.08995	-1.17255	9.496	2.51952	.90495	-1.29114

NAMELIST INPUT COMPLETED
GEOMETRY DATA DUMP, COMMON BLKGE1

6.8654	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4.4023	3.1234	0.0000	0.0000	0.0100	0.0000	0.0000	0.0000	0.0000	0.0000
1.5271	1.3474	0.0000	0.0190	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
.5061	.4454	.3009	.2418	.1319	.1200	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	4.1472	.3609	4.1472
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.5061
0.0000	0.0000	0.0000	0.0000	0.0300	1.0000	0.0000	0.0000	0.0000	.3009
.3100	5.1430	.9400	1.2308	2.0225	1.0000	0.0000	0.0000	0.0000	7.0000
4.1472	0.0000	0.0000	6.5411	0.0000	1.1969	0.0000	.3956	2.6042	0.0000
0.0000	.4454	.5061	.2418	.2198	0.0000	0.0000	.5061	0.0000	.3889
0.0000					-.1383	5.1343	2.6590	3.3047	.5549

Table 4 SAMPLE PROBLEM 3

```
WING 1     NASA TN D-5805      Q/C SWP = 25 DEG.
   1
NAMELIST INPUT
 $BODYS NBODYS=1,XCG=25.013,BLEN(1)=48.8,BWID(1)=4.9,BHGT(1)=4.9,
        BLNS(1)=14.0,BLBT(1)=10.0,BASE(1)=0.0$
 $NACEL NNACS=0$
 $WING  SREF=1.5,SPLAN=1.5,NPNLS=1,AR=6.0,TAPR=0.4,SWPLE=28.27,
        TW=5.,LAM(1)=0.0,TOC(1)=0.08,XLEW=20.587,YWW=2.45,YB=2.45,
        CBAR(1)=6.38,SWMT=25.0$
 $SURFS NSURFS=1,NHT=0,NVT=0$
 $SURV  NSURV=7,NHLSV=0,FMSURV(1)=0.225,0.456,0.665,0.717,0.77,
        0.822,0.883,ALT(1)=-1.5,-2.55,5*-3.5,DHSV(1)=7*0.0,
        SWPV(1)=7*28.27,CLLO(1)=7*0.0,CLHI(1)=7*1.0,NCLAS=21$
 $STOL  IHLS=0$
 $ADJUST IVAL(1)=20*0$
SAVE
WING 2  NASA TN D-5805  Q/C SWEEP = 35 DEG.
1 1
 $BODYS XCG=26.416$
 $WING  SWPLE=37.663,XLEW=20.6136,SWMT=35.$
SAVE
WING 3  NASA TN D-5805  Q/C SWEEP = 45 DEG.
1 1
 $BODYS XCG=28.2$
 $WING  SWPLE=47.01,XLEW=20.964,SWMT=45.$
SAVE
WING 4  NASA TN D-5805    Q/C SWEEP = 35 DEG.  T/C = 0.06
1 1 1
 $BODYS XCG=26.418$
 $WING  SWPLE=37.66,XLEW=20.6136,SWMT=35.,TOC(1)=0.06$
 $SURV  FMSURV(1)=0.292,.5,.702,.755,.807,.863,.932,
        ALT(1)=-1.8,-2.7,5*-3.6$
END
```

GENERAL DYNAMICS 6600 PROCEDURE RIT | CONVAIR AEROSPACE DIVISION PROBLEM 18628J-03 | FORT WORTH OPERATION 10/16/73 PAGE 0001

EMPIRICALLY BASED COMPUTER PROGRAM

TO PREDICT THE AERODYNAMIC CHARACTERISTICS

OF LARGE AIRCRAFT

WING 1 NASA TN D-5805 Q/C SWP = 25 DEG.

KPRINT(12) = 1
NAMELIST INPUT COMPLETED
GEOMETRY DATA DUMP, COMMON BLKG01

9.3592	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
.4934	.4364	.3761	.2469	.1361	.8604	0.0000	1.2202	.3761	1.2202
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.4934
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.3761
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0060	0.0000	6.0000
.4000	1.5000	.5306	.6429	1.9515	.6500	0.0000	.2857	1.2958	0.0000
1.2202	0.0000	0.0000	5.5344	0.0000	0.0000	0.0000	.4934	0.0000	.3761
0.0000	.4364	.4934	.2469	.2381	-.1117	0.0000	2.3201	0.0000	0.0000
0.0000									

CONFIGURATION SUMMARY

	LENGTH (FT.)	WETTED AREA (SQ. FT.)	FR OR T/C	INTERFERENCE FACTOR	MAX. T/C SWEEP (DEG.)
BODY NUMBER 1	.06667	4.36693	9.95918	1.00000	
WING PANEL NO. 1	.53167	2.46988	.08000		25.00001

6A SERIES AIRFOIL SECTION

GENERAL DYNAMICS CONVAIR AEROSPACE DIVISION FORT WORTH OPERATION
6600 PROCEDURE KIT PROBLEM 166280-03 10/16/73 PAGE 0003

MACH CRITICAL TABLE

CL	MACH CRITICAL
0.0000	.8675
.1000	.8516
.2000	.8351
.3000	.8165
.4000	.7932
.5000	.7658
.6000	.7342
.7000	.6986
.8000	.6578
.9000	.6159
1.0000	.5740

GENERAL DYNAMICS　　CONVAIR AEROSPACE DIVISION　　FORT WORTH OPERATION
6600 PROCEDURE R1T　　PROBLEM 186284-03　　10/16/73　PAGE 1364

WING 1　NASA TN D-5835　Q/C SWP = 25 DEG.
MACH NO. = .225　RN/FT = 1.5000E+06　L.E. SWEEP ANGLE = 28.27 DEG.　TAIL DEFL. (DH) = 0.00 DEG.

CL	TOTAL CD	CM	ALPHA	CD LIFT	CD RISE	CL AT DH=0	CD AT DH=0
0.000	.02049	0.00000	-.00	0.00000	0.00000	0.00000	.02049
.050	.02075	.00331	.61	.00026	0.00000	.05000	.02075
.100	.02153	.00661	1.22	.00104	0.00000	.10000	.02153
.150	.02283	.00992	1.82	.00234	0.00000	.15000	.02283
.200	.02465	.01323	2.43	.00416	0.00000	.20000	.02465
.250	.02699	.01653	3.04	.00651	0.00000	.25000	.02699
.300	.02985	.01984	3.65	.00937	0.00000	.30000	.02985
.350	.03324	.02315	4.26	.01275	0.00000	.35000	.03324
.400	.03714	.02645	4.86	.01665	0.00000	.40000	.03714
.450	.04156	.02976	5.47	.02108	0.00000	.45000	.04156
.500	.04651	.03307	6.08	.02602	0.00000	.50000	.04651
.550	.05263	.03596	6.74	.03214	0.00000	.55000	.05263
.600	.06338	.03285	7.62	.04429	0.00000	.60000	.06338
.650	.07893	.02449	8.76	.05844	0.00000	.65000	.07893
.700	.09926	-.00300	10.13	.07878	0.00000	.70000	.09926
.750	.12439	-.03951	11.77	.10390	0.00000	.75000	.12439
.800	.15430	-.09092	13.63	.13381	0.00000	.80000	.15430
.850	.18910	-.09865	20.75	.15851	0.00000	.85000	.18910
.900	.22849	-.10445	23.12	.20841	0.00000	.90000	.22849
.950	.27277	-.11026	25.73	.25229	0.00000	.95000	.27277
1.000	.32185	-.11606	26.59	.30136	0.00000	1.00000	.32185

CLA = .08223 PER DEG.　ALO = 0.00000 DEG.
K = .10428　DELCL = 0.00000　CLDB = .53272
CMO = 0.00000　CM/CL = .0661　CLMAX = .88178

DRAG BREAKDOWN -----		FUSLAGE	BODIES	NACELLES	WING	HORIZ TAIL	VERT TAIL	SURFACE
FRICTION	= .01678	.00928	0.00000	0.00000	.00750	0.00000	0.00000	0.00000
FORM	= .00175	.00079	0.00000	0.00000	.00096	0.00000	0.00000	0.00000
INTERF	= .00195	.00076	0.00000	0.00000	.00119	0.00000	0.00000	0.00000
WAVE	=0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
BASE	=0.00000	0.00000	0.00000					
CAMBER	=0.00000							
DRAG RISE	=0.00000							
MISC	=0.00000							

COMIN = .02349

4. PROGRAM AND SUBROUTINE DESCRIPTIONS

(arranged in alphabetical order)

This section contains a brief outline of the purpose and use of each program or subroutine.

Subroutine AALO

Purpose

To compute zero lift angle of attack. The zero-lift angle of attack is computed as the sum of the increments due to camber, twist and incidence.

Use

CALL AALO (SPEED)

where speed is the Mach number, the computed value of zero-lift angle is contained in COMMON BLKC01.

Subroutine ACCR

Purpose

ACCR computes the aerodynamic center at low lift and at stall for single panel wings. The subroutine also obtains the lift curve slope of the wing. The low lift aerodynamic center is obtained through use of triple-interpolation of the data presented in Figures 4.1.4.2-22 and -27 in the DATCOM.

Use

CALL ACCR (SPEED, AR, SWPLE, SWPMC, TR, SPLAN, TOC, TW,
FMCRO, XACR, CLAX, XACS)

where input is

SPEED	Mach number
AR	Aspect ratio of exposed wing
SWPLE	Leading-edge sweep
SWPMC	Mid-chord sweep
TR	Taper ratio

SPAN	Exposed planform area
TOC	Thickness ratio
TW	Type airfoil section indicator
FMCRO	Zero-lift critical Mach number for complete configuration

and output is

XACR	Low lift aerodynamic center referenced to leading-edge of exposed root chord
CLAX	Lift-curve slope of exposed planform
XACS	Aerodynamic center at stall

Subroutines called

TLNT
AER2
LNTP

Subroutine ADCL

Purpose

To compute the effect of camber on the displacement of the drag polar. For Mach numbers less than 1.0, the lift coefficient for minimum profile drag is computed; for Mach numbers greater than or equal to 1.0, the lift coefficient for minimum drag is computed.

Use

```
CALL ADCL (SPEED, CLOPT)
```

where speed is the Mach number and CLOPT is the C_L for minimum profile drag subsonically. Supersonically, CLOPT is the polar displacement C_L for $C_{D_{min}}$.

Subroutine ADJUST

Purpose

The subroutine adjusts an aerodynamic parameter, y_1, to a new value, y_2, by the equation

$$y_2 = y_1 \cdot V_M + V_A$$

where V_M and V_A are correlation multiplier and adder factors determined from input. The factors V_M and V_A are a function of either Mach number or C_L.

Use

CALL ADJUST (ID, ID2, XVAR, YVAR), where

ID	Parameter identification number for Mach number cases
ID2	Parameter identification number for C_L cases
XVAR	Value of Mach number if ID greater than 0, or value of C_L if ID = 0 and ID2 greater than 0
YVAR	Input value y_1 is changed to output value y_2

Subroutines called

LNTP

Subroutine AERA

Purpose

The subroutine calculates angle of attack for a given untrimmed C_L condition. For supersonic Mach numbers the linear relation

$$\alpha = C_L / C_{L_\alpha} + \alpha_{LO}$$

is used. For subsonic conditions the angle of attack is calculated by one of three different methods depending on whether the wing is a high aspect ratio, low aspect ratio or has a cranked leading edge. For low-aspect-ratio and cranked wings the effect of vortex lift is accounted for in the angle calculations.

Use

CALL AERA (SPEED, CL, ALPHA), where

SPEED	Mach number (I)
CL	Lift coefficient (I)
ALPHA	Angle of attack (O)

Subroutines called

DLNT
LNTP

Program AERO

Purpose

This program controls the sequence of calculations required to compute the trimmed lift, drag, moment, and angle of attack for a given set of conditions. The set of conditions are contained in COMMON BLKOV3 as one Mach number, altitude, trim indicator and untrimmed lift coefficient. The trim indicator is used to determine if a fixed horizontal tail setting is specified (in which case moment is calculated) or if the tail setting is calculated to produce a zero moment. The results of the aerodynamic calculations are contained in COMMON BLKC01.

The sequence of the calling of subroutines for aerodynamic calculations is controlled by the parameter JPASS, contained in COMMON BLKOV3, which is defined by program SURVEY. The parameter JPASS is used to prevent calling certain subroutines on repeat passes through AERO if the value they calculate remains fixed. An example would be where lift is the only change in the set of conditions received from COMMON BLKOV3; in this case it is unnecessary to recompute minimum drag for each CL.

Use

CALL OVERLAY (4HOVLY,3,3)

Subroutines called

ADJUST
AERA
AERA1 (second entry point in AERA)
AFTCD
ATMOS
CDDR
CDDR1 (second entry point in CDDR)
CDL1
CDL2
CDRG
CLBRK

(AERO continued)

Subroutines called

CLWBT
CMOW
DMIN
TDRG
WBAC

Subroutine AER2

Purpose

AER2 computes the lift-curve slope for a wing surface defined by the data in COMMON BLKCLA. The lift-curve slope is computed using a modified Polhamus expression in the subsonic range which is extended to match the two-dimensional linear-theory value at high supersonic Mach numbers.

Use

CALL AER2 (SPEED, CLA), where

SPEED	is the Mach number and
CLA	is the computed lift-curve value in per degree units

Subroutine AFTCD

Purpose

Subroutine AFTCD computes the drag increment, as a function of alpha, due to the fuselage aft-end upsweep.

Use

CALL AFTCD (ALPHA, CDAFT), where

ALPHA	is the angle of attack of the wing
CDAFT	is the aft-end drag increment due to upsweep angle

Subroutines called

DLNT

Subroutine ATMOS

Purpose

Given a geometric altitude H in feet the following quantities are computed:

T - Temperature in degrees Rankin

SIGMA - Ratio of density to that at sea level

D - Density in Lb-sec^2/ft^4

THETA - Ratio of temperature to that at sea level

DELTA - Ratio of pressure to that at sea level

A - Speed of sound in ft/sec

V - Viscosity coefficient in Lb-sec/ft^2

K - Error indicator

Data represents mean annual, mid-latitude, dry air conditions. Data below altitudes of -5000 meters or greater than 90 kilometers is invalid.

The equations and tables used in computation are essentially the same as those used in computing U.S. Standard Atmosphere, 1962, and this routine maintains the same degree of accuracy as the tables in U.S. Standard Atmosphere, 1962.

Use

CALL ATMOS (H, T, SIGMA, D, THETA, DELTA, A, V, K)

Reference

U.S. Standard Atmosphere, 1962; Government Printing Office

Subroutine BDRG

Purpose

To compute the base drag of an arbitrary body. The subroutine uses an empirical equation to determine base drag.

Use

CALL BDRG (SPEED, AB, SREF, CBD), where

SPEED	Mach number (I)
AB	Base area (I)
SREF	Reference area (I)
CDB	Base drag coefficient (O)

BLOCK DATA

Purpose

To define data in COMMON blocks which represent tables and charts that are used in various subroutines.

Subroutine CDDR

Purpose

Subroutine CDDR calculates the drag rise along with defining the two limit Mach numbers and the lift-curve slope at those Mach numbers. The limit Mach numbers and associated lift-curves are used later in subroutine CDL1 to define the polar shape in the transonic region.

CDDR has two entry points, on the first pass through the constants in the drag rise equation are computed along with the two limit Mach numbers and their associated lift-curves. For subsequent passes only the drag rise is computed.

Use

```
CALL CDDR  (CL, XMACH, RNOFT, CDR)
CALL CDDR1
```

Subroutines called

CLWBT
FDRG
LNTP
WDRG

Subroutine CDL1

Purpose

This subroutine calculates the constants which are used by subroutine CDL2 to determine the drag polar.

Use

CALL CDL1 (SPEED, RNOFT, FK, DELCL, PRIMEK, AKD, AKB), where

SPEED	Mach number (I)
RNOFT	Reynolds no./foot (I)
FK	Polar shape factor below polar break lift coefficient (O)
DELCL	Polar lift displacement (O)
PRIMEK	Additional drag factor for drag polar above polar break (O)
AKD	Theoretical drag-due-to-lift factor (O)
AKB	Separation drag factor used to calculate drag polar above separation lift coefficient (O)

Subroutines called

ADCL
DLNT
KGIN
LNTP

Subroutine CDL2

Purpose

This subroutine computes the drag-due-to-lift using the polar shape factors determined by subroutine CDL1 and the polar break and separation lift coefficients determined by subroutine CLBRK.

Use

CALL CDL2 (SPEED, CL, AEROK, DELCL, PRIMEK, AKD, AKB, CDL), where

SPEED	Mach number (I)
CL	Lift coefficient (I)
AEROK	Polar shape factor below polar break lift coefficient (I)
DELCL	Polar lift displacement (I)
PRIMEK	Additional drag factor for drag polar above polar break (I)
AKD	Theoretical drag-due-to-lift factor (I)
AKB	Separation drag factor (I)
CDL	Drag due to lift (O)

Subroutine CDRG

Purpose

This subroutine calculates the drag increment due to wing camber. Wing camber causes a lift displacement in the drag polar; this displacement lift increment is related to the difference between the minimum profile drag and the minimum drag of the polar.

Use

CALL CDRG (SPEED, AEROK, DELCL, CDC), where

SPEED	Mach number (I)
AEROK	Polar shape factor (I)
DELCL	Polar lift displacement (I)
CDC	Camber drag (O)

Subroutine CDWN

Purpose

This subroutine calculates the nose wave drag of body and nacelle components.

Use

CALL CDWN (AMAX, XLNOS, RIN, CDW), where

AMAX	Maximum cross-sectional area (I)
XLNOS	Length of nose (I)
RIN	Radius of inlet area (I)
CDW	Wave drag of component based on maximum cross-sectional area (O)

Subroutine called

DLNT

Subroutine CDWT

Purpose

This subroutine calculates the boattail wave drag of body and nacelle components.

Use

CALL CDWT (AMAX, XLAFT, REX, CDW), where

AMAX	Maximum cross-sectional area (I)
XLAFT	Length of boattail (I)
REX	Exit or base area (I)
CDW	Wave drag of component based on maximum cross-sectional area (O)

Subroutine called

LNTP

Subroutine CDWW

Purpose

This subroutine calculates the wave drag for airfoil surface components.

Use

CALL CDWW (CDOSR), where CDOSR is the wing wave drag based on the configuration reference area.

Also the data in COMMON BLKWPD must be defined prior to calling CDWW. The data in BLKWPD is

AR	Aspect ratio of surface based on exposed planform area
ZLAM	Taper ratio of exposed planform
ZLE	Leading-edge sweep
ZTE	Trailing-edge sweep
ZM	Mach number
SOSR	Exposed area to reference area ratio
TYPE	Airfoil type indicator
CAM	Section camber
TOC	Section thickness to chord ratio

Subroutine CFEQ

Purpose

This subroutine calculates the flat-plate skin friction coefficient using the White-Christoph technique for turbulent

flow, the Blasius relation for laminar flow, and a momentum thickness matching technique for partial laminar-turbulent flow.

Use

CALL CFEQ (RNOFT, ZMACH, CBAR, XTR, CF), where

RNOFT	Reynolds number/foot
ZMACH	Mach number
CBAR	Length
XTR	Distance along CBAR where transition occurs
CF	Skin friction coefficient

Subroutine CLBRK

Purpose

Subroutine CLBRK calculates the lift coefficients for polar break, separation drag onset, and maximum lift. The subroutine also calculates other aerodynamic parameters used in subroutine AERA to compute angle of attack as a function of C_L.

Use

CALL CLBRK (SPEED, RE, RNOFT), where

SPEED	Mach number (I)
RE	Reynolds number parameter (I)
RNOFT	Reynolds number/foot (I)

The output, contained in COMMON BLKC01, is defined as,

CLPB	Polar break lift coefficient
CLDB	Separation lift coefficient
CLMAX	Maximum lift coefficient
ABRK	Angle of attack for polar break
AMAX	Angle of attack at CLMAX
DAMAX	Increment in angle of attack between AMAX and a linear value of alpha at CLMAX
DEL	Tail lift increment to CLMAX
CLS	Lift coefficient where CL versus alpha becomes nonlinear
ARLO	Aspect ratio limit between low AR and high AR calculations

Subroutines called

DLNT
LNTP

Subroutine CLWBT

Purpose

CLWBT controls the sequence of calculations that compute the total wing-body-tail lift curve slope, zero lift angle of attack, and the factors used to compute drag and lift increments due to a horizontal tail deflection.

Use

CALL CLWBT (SPEED), where SPEED is the Mach number, the output is contained in COMMON BLKC01.

Subroutines called

AALO
ADJUST
AER2
TAIL

Subroutine CMOW

Purpose

Subroutine CMOW computes the moment at zero-lift for the wing-body configuration.

Use

CALL CMOW (SPEED, CMO), where

SPEED	Mach number (I)
CMO	Wing-body C_{m_O}

Subroutine called

TLNT

Subroutine CONV

Purpose

CONV converts input data to feet and radian units for the namelist input program NINPT. The subroutine makes the conversion only to the variables that are redefined in the NINPT program.

Use

CALL CONV
CALL CONV1

Subroutine CPUOV

Purpose

This subroutine calculates the pressure coefficient, C_P at x/c for an infinite sheared wing. The C_P is obtained by solving Equations (93) and (90) in the Royal Aero. Soc. TDM-6312. The arrays corresponding to $S^{(1)}(x)$, $S^{(2)}(x)$, $S^{(3)}(x)$, $S^{(4)}(x)$, and $S^{(5)}(x)$ in Equation (93) are obtained from COMMON BLKCPI and were defined in subroutine CPZT.

Use

CALL CPUOV (S, A, SWP, IV, CPI, CP, XM), where

S	Sign indicator (+1 for upper surface C_P, and -1 for lower surface C_P) (I)
A	Angle of attack (I)
SWP	Sweep angle (I)
IV	Control point, $x/c = \frac{1}{2}(1 - \cos(\frac{IV\,\pi}{32}))$ (I)
CPI	Incompressible C_P (O)
CP	Compressible value of C_P (O)
XM	Mach number (I)

Subroutine CPZT

Purpose

This subroutine computes the critical Mach number using the local Mach number normal to the isobar in the mid-span region

of the wing. The mid-chord sweep and the aspect ratio of the wing are used to define an effective isobar sweep. The subroutine obtains the airfoil geometry from subroutine SECT and the pressure distribution around the airfoil from subroutine CPUOV. The subroutine then uses the incompressible pressure at the crest of the airfoil and isentropic flow relationships to calculate the critical Mach number on the wing. The predicted wing critical Mach number is prevented from exceeding the critical Mach number of the fuselage which is calculated as a function of fuselage fineness ratio.

Use

CALL CPZT (ID, XMACH, TOC, CLD, SWEEP), where

ID	Type identification of the airfoil (I)
XMACH	Mach number for compressible Cp solutions (I) (set equal to 0.6 in Program MCRIT)
TOC	Airfoil section thickness ratio (I)
CLD	Airfoil section camber (I)
SWEEP	Mid-chord sweep at mid-semi-span of the wing (I)

The output which consists of a series of critical Mach number and lift combinations, ceasing from a six-degree angle of attack are contained in COMMON BLKA05.

Subroutines called

CPUOV
LNTP
SECT

Subroutine DAERO

Purpose

This subroutine computes the lift, moment and drag in free air and in ground effect of the high-lift system. Subroutine DAERO uses the incremental effect of flap and leading-edge device computed in subroutines DSET and MSET along with the clean airplane aerodynamics computed by program AERO to compute the total lift, drag and moment versus angle of attack in both free air and ground effect.

Use

CALL DAERO (ALPHA, CL, CD, CM, AGRD, CLG, CDG, CMG, H, DF), where

ALPHA	Angle of attack (I)
CL	Lift coefficient (O)
CD	Drag coefficient (O)
CM	Moment coefficient (O)
AGRD	Angle of attack in ground effect (O)
CLG	Lift coefficient in ground effect (O)
CDG	Drag coefficient in ground effect (O)
CMG	Moment coefficient in ground effect (O)
H	Height of wing $\bar{c}/4$ point above ground (I)
DF	Flap deflection (I)

Subroutines called

DLNT
LNTP

Function DLNT

Purpose

DLNT is a two-dimensional, nth-order Lagrangian interpolation procedure.

Use

CALL DLNT (XBAR, YBAR, X, Y, F, NX, NY, NXMAX, LOX, LOY), where

XBAR	The X value at which a value of the function is to be interpolated (I)
YBAR	The Y value at which a value of the function is to be interpolated (I)
X	The array of X values (I)
Y	The array of Y values (I)
F	The values of the function f(x,y) (I)
NX	The size of the X array and the F array in the X direction (I)
NY	The size of the Y array and the F array in the Y direction (I)
NXMAX	The dimension of the F array in the X direction in the calling routine (I)
LOX, LOY	Number of points to be used in the X and Y directions, respectively, in the interpolations: 1 for step, 2 for linear, 3 for parabolic, 4 for cubic, etc. (I)

Subroutine called

LNTP

Subroutine DMIN

Purpose

Subroutine DMIN controls the sequence of calculations necessary to compute minimum drag.

Use

CALL DMIN (SPEED, RNOFT, CDMIN), where

SPEED	Mach number (I)
RNOFT	Reynolds number/ft (I)
CDMIN	Minimum drag (O)

Subroutines called

ADJUST
BDRG
FDRG
WDRG

Subroutine DSET

Purpose

Subroutine DSET computes the incremental lift and drag of flaps and leading-edge devices. This subroutine uses the two-dimensional section data calculated in subroutine SSET along with the geometry of the high lift system in order to compute the incremental effect on the airplane.

Use

CALL DSET (BFI, BFO, BSI, BSO, CF), where

BFI	Inboard span station of flap (I)
BFO	Outboard span station of flap (I)
BSI	Inboard span station of leading-edge device (I)
BSO	Outboard span station of leading-edge device (I)
CF	Chord ratio of trailing-edge flap (I)

The output is contained in COMMON BLKCO2 where

DCLOF	Increment in lift at zero angle of attack due to flap deflection
DCLOS	Increment in lift at zero angle of attack due to leading-edge device
DCLMF	Increment in maximum lift due to flap deflection
DCLMS	Increment in maximum lift due to leading-edge device
DCDMIN	Increment in minimum drag due to high-lift system
DCLF	Displacement in drag polar due to flap deflection
DCLS	Displacement in drag polar due to leading-edge device
SPLANX	Planform area with high-lift system deployed
RCLA	Ratio of lift curve slope with high-lift system to clean airplane lift-curve slope.

Subroutines called

DLNT
LNTP

Subroutine FDRG

Purpose

This subroutine calculates friction, form and interference drag for all the components on the airplane.

Use

CALL FDRG (SPEED, RNOFT), where

SPEED	Mach number (I)
RNOFT	Reynolds number/ft (I)

The output is contained in COMMON BLKC01.

Subroutines called

CFEQ
FFACT
IFACT

Subroutine FFACT

Purpose

Subroutine FFACT computes the form factors for each component.

Use

CALL FFACT (ID, GEOM, TYP, CLD, SPEED, CRITM, FF), where

ID	Identification for bodies, nacelles, or surface components (I)
GEOM	Fineness ratio for bodies and nacelle components; or thickness ratio for surface components (I)
TYP	Airfoil type identification number (I)
CLD	Airfoil camber (I)
SPEED	Mach number (I)
CRITM	Critical Mach number of configuration (I)
FF	Form factor (O)

Program GEOM

Purpose

Program GEOM computes many of the geometric parameters that are used in aerodynamic calculations. The geometry that was read in as input is used to calculate the additional geometry needed for aerodynamic calculations. The input geometry is contained in COMMON's BLKA01, BLKA02, and BLKA03 and the geometry calculated in program GEOM is output in COMMON BLKG01.

Use

CALL OVERLAY (4HOVLY, 2, 0)

Subroutine IFACT

Purpose

Subroutine IFACT computes the interference factors for each component.

Use

CALL IFACT (ID, PARM, CRITM, SPEED, FI), where

ID	Identification for bodies or surface components (I)
PARM	Fuselage Reynolds number for bodies, or maximum thickness sweep for surface components (I)
CRITM	Critical Mach number of configuration (I)
SPEED	Mach number (I)
FI	Interference factor (O)

Subroutine called

DLNT

Program INPT

Purpose

The configuration geometry and the aerodynamic conditions to be run are read in using formatted input statements. The program also converts the input data from inch and degree units to feet and radians.

Use

CALL OVERLAY (4HOVLY, 1, 1)

Subroutine KGIN

Purpose

This subroutine computes the polar shape factor and polar displacement for a drag polar with drag rise added. A least-squares, second-degree curve is fitted to the drag polar with drag rise. This is needed by subroutine CDL1 in order to interpolate the polar shape in the transonic region between M_{L1} and M_{L2}.

Use

CALL KGIN (CLDB, AKIN, DECLIN, SPEED, AKOUT, DCLOUT), where

CLDB	Upper C_L limit for polar calculation (I)
AKIN	Polar shape factor without drag rise (I)
DECLIN	Polar displacement C_L without drag rise (I)

SPEED	Mach number (I)
AKOUT	Equivalent polar shape factor with drag rise (O)
DCLOUT	Equivalent polar displacement with drag rise (O)

Subroutines called

CDDR1
LSPCF

Subroutine LNTP

Purpose

LNTP is a one-dimensional, nth-order Lagrangian interpolation procedure.

Use

CALL LNTP (XBAR, YBAR, X, Y, M, NO), where

XBAR	The abscissa value at which an ordinate is to be interpolated (I)
YBAR	The interpolated ordinate (O)
X	The array of abscissas (I)
Y	The array of ordinates (I)
M	The size of the arrays (I)
NO	The number of points to be used in the interpolation. NO=1 for step, NO=2 for linear, NO=3 for parabolic, NO=4 for cubic, etc. (I)

Program LSHL

Purpose

LSHL controls the sequence of calculations to produce lift, moment, and drag variations with angle of attack for each low-speed, high-lift survey condition specified by the input. For each high-lift survey, the program computes the incremental effect for an inboard and an outboard high-lift system segment. The program then sets up a DO LOOP to calculate the total lift, drag and moment for changing angle of attack up to C_{LMAX}.

Use

CALL OVERLAY (4HOVLY, 3, 4)

Subroutines called

DAERO
DSET
MSET
SSET

Program MCRIT

Purpose

Program MCRIT constructs a table of critical Mach number versus C_L from either an input table or by using an empirical method.

Use

CALL OVERLAY (4HOVLY, 3, 2)

Subroutines called

ADJUST
CPZT
LNTP

Program MRIT (MAIN PROGRAM)

Purpose

This program controls the logic of the calling of the three primary overlays. The lengths of the principal COMMON blocks used in all three overlays are also specified in this program.

Programs called

OVERLAY(4HOVLY, 1, 0)	(XINPT)
OVERLAY(4HOVLY, 2, 0)	(GEOM)
OVERLAY(4HOVLY, 3, 0)	(SURVEY)

Subroutine MSET

Purpose

Subroutine MSET computes the incremental moment at zero lift of flaps and leading-edge devices. This subroutine uses the two-dimensional section data calculated in Subroutine SSET along with the geometry of the high-lift system in order to compute the incremental effect on the airplane.

Use

CALL MSET (BFI, BFO, BSI, BSO), where

BFI	Inboard span station of flap (I)
BFO	Outboard span station of flap (I)
BSI	Inboard span station of leading-edge device (I)
BSO	Outboard span station of leading-edge device (I)

The output is contained in COMMON BLKO2 where,

DCMO	Increment in C_{m_0} due to high-lift system
CDMCL	Increment in slope of the moment versus lift curve

Subroutines called

DLNT

Subroutine MTXEQ

Purpose

MTXEQ solves a set of simultaneous equations. This subroutine will solve the matrix equation AX = B for the unknown matrix X. The dimensions of the matrices must be:

A: NxN B: NxM

where M is the number of B vectors to be transformed into X vectors. At the same time, this subroutine computes a scaled version of the determinant of the matrix A.

The solution of the matrix equation AX = B is accomplished by upper triangularizing the A matrix using a maximum row element as a pivot point for each stage in the reduction of the matrix A.

This entails searching in the first column of the reduced (N-K) x (N-K) A matrix for the element whose absolute value is the largest. A row interchange is then performed to bring this element into the A_{KK} position. After completion of the triangularization, back substitution is used to obtain the X matrix.

Use

CALL MTXEQ (A, X, B, N, K), where

A	The array containing the elements of the matrix A. Array A must have dimensions NxN.
X	The array containing the elements of the matrix X. Array X must have dimensions NxK.
B	The array containing the elements of the matrix B. Array B must have dimensions NxK.
N	Number of columns and rows in the matrix A.
K	Number of columns in the matrix B and X.

Program NINPT

Purpose

The configuration geometry and the aerodynamic conditions to be run are read in using NAMELIST input statements. The program calls subroutine CONV to convert the input data to feet and radians units. For cases which are to be rerun with only a slight change in input from the previous case, an indicator is used to read only certain sets of namelist data.

Use

CALL OVERLAY (4HOVLY, 1, 2)

Subroutines called

CONV
CONV1 (second entry point in CONV)

Subroutine PLSQ

Purpose

To fit a polynomial of degree K to the set of points (X_i, Y_i) for $i = 1, \ldots, N$, by the method of least squares.

The method of solution is as follows:

Given a set of N points, coefficients are found such that:

$$Y = C(K+1) + C(K)\cdot X + C(K-1)\cdot X^2 + \ldots$$

$$+ C(\ )\cdot X^{K-1} + C(1)\cdot X^K$$

is the best least-square fit to Y_i for an K<u>th</u> degree fit.

PLSQ solves the regression matrix equation by calling MTXEQ.

Use

CALL PLSQ (X, Y, N, K, C, LIST, EMAX, ERMS, EMEQ), where

X	Array of N independent variables
Y	Array of N dependent variables
N	Number of variables
K	Degree of the least-squares polynomial
C	Array of coefficients, high order to low order, of the least-squares polynomial
LIST	Error analysis output indicator =0 suppresses error analysis printout =1 prints error analysis
EMAX	Maximum absolute error obtained by using the least-square fit polynomial to approximate the dependent variable
ERMS	RMS error obtained by using the least-square fit polynomial
EMEQ	Maximum derivation from unity in the linear system check solution

Subroutine called

MTXEQ

Subroutine SECT

Purpose

Subroutine SECT calculates the thickness and camber airfoil ordinates which are used in subroutine CPZT to calculate pressure distributions. SECT can calculate, using internal methods, the section data for the standard NACA 6-series and 4-digit airfoils along with a biconvex and Whitcomb's supercritical airfoil. The subroutine can also obtain the airfoil ordinates at the control points, x/c, needed for pressure solutions by interpolation on a table of input ordinates.

Use

CALL SECT (ID, TOC, CLD), where

ID	Airfoil section identification number (I)
TOC	Airfoil thickness ratio (I)
CLD	Airfoil camber (I)

The output is contained in COMMON BLKR07.

Subroutine called

LNTP

Subroutine SETUP

Purpose

Places initial values of 0.0 in the input and geometry common blocks. This is done to prevent indeterminate values to be used in program calculations if input data did not specify values.

Use

CALL SETUP

Subroutine SSET

Purpose

Subroutine SSET computes the two-dimensional section data used in subroutines DSET and MSET to compute the incremental lift, drag and moment due to a high-lift system. SSET either computes the two-dimensional increments using a empirical modification of thin-airfoil theory, or SSET obtains the two-dimensional increments from input data if available.

Use

CALL SSET (DF, DS, CFOC, CSOC, OTE, ROT), where

DF	An array containing the flap deflection of each element of the trailing-edge flap (I)
DS	Deflection of the leading-edge device (I)
CFOC	An array containing the chord length ratios of each element of the trailing-edge flap (I)
CSOC	Chord length ratio of the leading-edge device (I)
OTE	Airfoil trailing-edge angle (I)
ROT	Airfoil leading-edge radius divided by thickness (I)

The output is contained in COMMON BLKC01, where

DC1F	Two-dimensional increment in lift at zero angle of attack due to flap deflection
DC1MF	Two-dimensional increment in maximum lift due to flap deflection
DCDFS	Increment in profile drag due to flap deflection
DCMFS	Two-dimensional increment in momentum at zero alpha due to flap deflection
DC1S	Two-dimensional increment in lift at zero angle of attack due to leading-edge device
DC1MS	Two-dimensional increment in maximum lift due to leading-edge device
DCDSS	Increment in profile drag due to leading-edge device

DCDMSS Two-dimensional increment in moment at zero alpha due to leading-edge device

Subroutines called

DLNT
LNTP

Program SURVEY

Purpose

SURVEY controls the sequence of calculations to produce a lift, moment and drag variation for each high-speed survey condition specified by the input.

For variable sweep configurations the program will first call program VGEOM with the wing leading-edge sweep set at the forward position and then recall program VGEOM with the sweep set at the aft position. This is done in order to setup program VGEOM for geometry calculations at any arbitrary sweep position. SURVEY then enters a DO LOOP where the high sweep survey conditions are set up. SURVEY calls VGEOM and MCRIT to recalculate the geometry and the configuration critical Mach number each time the leading-edge sweep is changed in a survey. SURVEY then enters an inner DO LOOP where a sequence of untrimmed C_L are generated and program AERO is called to obtain the trimmed lift, moment and drag. SURVEY then prints out the results.

If a series of low-speed, high-lift surveys were specified by the input, SURVEY then calls program LSHL to perform those calculations.

Use

CALL OVERLAY (4HOVLY, 3, 0)

Programs called

OVERLAY(4HOVLY, 3, 1)	(VGEOM)
OVERLAY(4HOVLY, 3, 2)	(MCRIT)
OVERLAY(4HOVLY, 3, 3)	(AERO)
OVERLAY(4HOVLY, 3, 4)	(LSHL)

Subroutine TAIL

Purpose

Subroutine TAIL computes the lift curve slope contribution of the tail along with factors used to compute lift and drag increments due to a horizontal tail deflection. These factors are computed by first solving for the downwash, dynamic-pressure, exposed area lift-curve slope, carry-over lift factors and induced drag for the tail.

Use

CALL TAIL (SPEED), where SPEED is the input Mach number.

The output, contained in COMMON BLKC01, is

CLAT	Lift-curve slope contribution of the tail
A	Trim drag factor
B	Trim drag factor
AOH	Angle of zero lift of the tail
CLDH	Change in lift due to tail deflection factor
DEDA	Change in downwash per change in angle of attack

Subroutines called

AER2
LNTP

Subroutine TDRG

Purpose

Subroutine TDRG calculates moment using the wing-body C_{m_o} and aerodynamic center along with the tail lift and moment arm. Depending upon user options the moment can be calculated at a fixed horizontal tail setting, or a tail setting can be computed which will trim out the moment. Lift and drag increments due to tail deflection are also computed.

Use

CALL TDRG (ITRIM, SPEED, DCLT, DCDT), where

ITRIM	Trim indicator, =0 if tail setting specified by input, =1 if tail setting is to be computed to trim out moment
SPEED	Mach number
DCLT	Increment in lift due to tail deflection relative to zero tail setting
DCDT	Increment in drag due to tail deflection relative to zero tail setting

Subroutine TLNT

Purpose

TLNT is a triple-linear interpolation procedure.

Use

CALL TLNT (XBAR, YBAR, ZBAR, FBAR, X, Y, Z, F, NX, NY, NZ, NXMAX, NYMAX), where

XBAR	The X value at which a value of the function is to be interpolated (I)
YBAR	The Y value at which a value of the function is to be interpolated (I)
ZBAR	The Z value at which a value of the function is to be interpolated (I)
FBAR	The interpolated value of the function F(X,Y,X) (I)
X	The array of X values (I)
Y	The array of Y values (I)
Z	The array of Z values (I)
F	The three-dimension array F values (I)
NX	The size of the X array and the F array in the X direction (I)
NY	The size of the Y array and the F array in the Y direction (I)
NZ	The size of the Z array and the F array in the Z direction (I)
NXMAX	The dimension of the F array in the X direction in the calling routine (I)
NYMAX	The dimension of the F array in the Y direction in the calling routine (I)

Subroutines called

DLNT, LNTP

Program VGEOM

Purpose

Program VGEOM computes the geometry parameters that vary with wing sweep for variable-sweep configurations. The program is first called by program SURVEY at the forward and most aft sweep positions in order to set up VGEOM for any arbitrary sweep calculation.

Use

CALL OVERLAY (4HOVLY, 3, 1)

Subroutine WBAC

Purpose

Subroutine WBAC computes the aerodynamic center location for wing-body configurations. The subroutine calculates the aerodynamic center of the wing carry-over lift on the body, and the aerodynamic center of the forebody. A composite aerodynamic center is then computed using the aerodynamic center and lift curve slope of each component.

Use

C..LL WBAC (SPEED, XACR), where

SPEED	Mach number (I)
XACR	Aerodynamic center of the wing-body configuration reference to the leading edge of the exposed root chord

Subroutines called

ACCR
DLNT
LNTP

Subroutine WDRG

Purpose

This subroutine calculates the wave drag for all the components on the airplane.

Use

CALL WDRG (FMACH), where

FMACH Mach number

The output is contained in COMMON BLKC01.

Subroutines called

CDWN
CDWT
CDWW

Program XINPT

Purpose

XINPT controls the sequence of input for new problems or allows the previous problem to be repeated with only a small change in input. For each new problem the title card, print dump card, and input control card are read from the input file. The subroutine SETUP is then called to zero out all the input common blocks, then depending upon the input control card either of the secondary overlay programs INPT or NINPT are then called to continue reading the input file. After the input file is read for a given problem, control is then transferred back to MRIT where the solution to the problem is obtained. After each solution is obtained control is then transferred back to XINPT and the repeat problem control card is then read. If end-of-file on tape 5 is read the program is terminated. If a perturbation of the previous problem is indicated, NINPT is called where the revised input is then read.

Use

CALL OVERLAY (4HOVLY, 1, 0)

Programs called

SETUP	
OVERLAY(4HOVLY, 1, 1)	(INPT)
OVERLAY(4HOVLY, 1, 2)	(NINPT)
EXIT	

5. PROGRAM LISTING

```
      OVERLAY (OVLY,0,0)                                                 R1T0001
      PROGRAM MR1T(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)                R1T0002
C     LARGE AIRCRAFT AERODYNAMIC PREDICTION PROCEDURE                    R1T0003
C                                                                        R1T0004
      COMMON /BLKA01/ N4(7)                                              R1T0005
      COMMON /BLKA02/ A02(433)                                           R1T0006
      COMMON /BLKA03/ A03(20)                                            R1T0007
      COMMON /BLKA04/ N3(3), A04(78)                                     R1T0008
      COMMON /BLKA05/ A05(131), NMCR, NXSET                              R1T0009
      COMMON /BLKSUR/ N1(42), SUR(370)                                   R1T0010
      COMMON /BLKHLS/ N2, HLS(70)                                        R1T0011
      COMMON /BLKG01/ G1(200)                                            R1T0012
      COMMON /BLKPRT/ KPRINT(50)                                         R1T0013
      COMMON /BLKADJ/ ADJ(322)                                           R1T0014
      COMMON /BLKTIL/ TITLE(6)                                           R1T0015
      COMMON /BLKOV1/ IJ                                                 R1T0016
C                                                                        R1T0017
   15 IJ     = 0                                                         R1T0021
   20 CALL OVERLAY(4HOVLY,1,0)                                           R1T0022
C                                                                        R1T0023
      CALL OVERLAY(4HOVLY,2,0)                                           R1T0024
      CALL OVERLAY(4HOVLY,3,0)                                           R1T0025
C                                                                        R1T0026
      GO TO 20                                                           R1T0027
C                                                                        R1T0028
      END                                                                R1T0029
```

CC = 00026

```
      SUBROUTINE LNTP(XBAR,YBAR,X,Y,M,NO)
C
C     LAGRANGIAN INTERPOLATION
C           1) INCREASING OR DECREASING X ARRAY
C           2) LINEAR EXTRAPOLATION ONLY
C           3) NO = NUMBER OF POINTS USED IN INTERPOLATION
C           4) M  = TOTAL NUMBER OF POINTS IN X ARRAY
C           5) IF(NO.LE.0.OR.M.LE.0) YBAR = 0.0
C           6) IF(M.EQ.1.AND.NO.GT.0) YBAR=Y(1)
C           7) IF(NO.EQ.1.AND.M.GT.0) YBAR=Y(NEAREST X(I))
C           8) IF(NO.GT.M) INTERPOLATION AS IF NO=M
C
      DIMENSION  X(M) , Y(M)
C
      YBAR=0.0
      N=NO
      IF(N.GT.M) N=M
      IF(M.LE.0.OR.N.LE.0) RETURN
      IF(M.GT.1) GO TO 10
      YBAR=Y(1)
      RETURN
C
   10 MDC=0
      IF(X(M).LT.X(1)) MDC=1
      DO 20 I=2,M
      I1=I-1+MDC
      I2=I-MDC
      IF(XBAR.GE.X(I1).AND.XBAR.LE.X(I2)) GO TO 40
   20 CONTINUE
      I1=1
      IF((MDC.EQ.0.AND.XBAR.GE.X(M)).OR.(MDC.NE.0.AND.XBAR.LE.X(M)))
     1   I1=M-1
      I2=I1+1
      GO TO 45
C
   40 IF(N.GT.2) GO TO 60
   45 IF(N.LT.2) GO TO 55
   50 YBAR=Y(I1)+(XBAR-X(I1))*(Y(I2)-Y(I1))/(X(I2)-X(I1))
      RETURN
C
   55 YBAR=Y(I1)
      IF(ABS(XBAR-X(I2)).LT.ABS(XBAR-X(I1))) YBAR=Y(I2)
      RETURN
C
   60 J=I-N/2
      IF(I.LE.N/2+1) J=1
      IF(I.GT.M-N/2) J=M-N+1
      NT=N+J-1
C
      DO 80 I=J,NT
      ELL=1.0
      DO 70 K=J,NT
      IF(K.NE.I) ELL=ELL*(XBAR-X(K))/(X(I)-X(K))
   70 CONTINUE
   80 YBAR=YBAR+ELL*Y(I)
      RETURN
```

```
C                                                                     R1T0087
      END                                                             R1T0088
                                                         CC = 00058
```

```
      FUNCTION DLNT (XBAR,YBAR,X,Y,F,NX,NY,NXMAX,LOX,LOY)
C
C     DOUBLE LAGRANGE INTERPOLATION
C
      DIMENSION X(1),Y(1),F(NXMAX,1),FT(25)
C
      FBAR=0.0
      IF(LOY.GT.25) LOY=25
      LLX=LOX
      IF(LOX.GT.NX) LLX=NX
      IF(NY.GT.1) GO TO 10
      CALL LNTP (XBAR,FBAR,X,F(1,1),NX,LLX)
      GO TO 30
C
   10 LLY=LOY
      IF(LOY.GT.NY) LLY=NY
      DO 20 I=1,NY
      CALL LNTP (XBAR,FT(I),X,F(1,I),NX,LLX)
   20 CONTINUE
      CALL LNTP (YBAR,FBAR,Y,FT,NY,LLY)
C
   30 DLNT=FBAR
C
      RETURN
      END
```

CC = 00025

```
      SUBROUTINE TLNT(XBAR,YBAR,ZBAR,FBAR,X,Y,Z,F,NX,NY,NZ,NXMAX,NYMAX)
C
C         TRIPLE LINEAR INTERPOLATOR
C
      DIMENSION X(1),Y(1),Z(1),F(NXMAX,NYMAX,1),FT(25)
C
      DO 10  I=1,NZ
C
   10 FT(I)  = DLNT(XBAR,YBAR,X,Y,F(1,1,I),NX,NY,NXMAX,2,2)
C
      CALL LNTP(ZBAR,FBAR,Z,FT,NZ,2)
C
      RETURN
      END
```

CC = 00014

```
      OVERLAY(1,0)                                                      R1T0131
      PROGRAM XINPT                                                     R1T0132
C                                                                       R1T0133
      COMMON /BLKPRT/ KPRINT(50)                                        R1T0134
      COMMON /BLKTIL/ TITLE(6)                                          R1T0135
      COMMON /BLKOV1/ IJ                                                R1T0136
C                                                                       R1T0137
      DIMENSION W(4), IS(12)                                            R1T0138
      DATA  W  / 4HFORM, 4HNAME, 4HSAVE, 4HEND /                        R1T0139
C                                                                       R1T0140
      IF( IJ.GE.1 ) GO TO 70                                            R1T0141
   20 CONTINUE                                                          R1T0142
C                                                                       R1T0143
C                                                                       R1T0145
      READ(5,1000) (TITLE(I), I = 1,6 )                                 R1T0146
      IF (EOF(5).NE.0) CALL EXIT                                        R1T0147
      WRITE(6,2000) (TITLE(I), I= 1,6 )                                 R1T0148
C                                                                       R1T0149
      READ(5,1001) (KPRINT(I), I=1,50)                                  R1T0150
C                                                                       R1T0151
      DO 50 I = 1, 50                                                   R1T0152
      IF( KPRINT(I).GT.0 ) WRITE(6,2001) I, KPRINT(I)                   R1T0153
   50 CONTINUE                                                          R1T0154
      CALL SETUP                                                        R1T0155
C                                                                       R1T0156
      IJ     = 1                                                        R1T0157
   70 READ(5,1002) WORD                                                 R1T0158
      IF( EOF(5).NE.0 ) CALL EXIT                                       R1T0159
      IF( WORD.EQ.W(1) ) CALL OVERLAY(4HOVLY,1,1)                       R1T0160
      IF( WORD.EQ.W(2) ) IJ = 1                                         R1T0161
      IF( WORD.EQ.W(2) ) CALL OVERLAY(4HOVLY,1,2)                       R1T0162
      IF( WORD.EQ.W(3) ) IJ = 2                                         R1T0163
      IF( WORD.EQ.W(3) ) CALL OVERLAY(4HOVLY,1,2)                       R1T0164
      IF( WORD.EQ.W(4) ) GO TO 20                                       R1T0165
C                                                                       R1T0166
 1000 FORMAT( 6A10 )                                                    R1T0167
 1001 FORMAT( 50I1 )                                                    R1T0168
 1002 FORMAT( A4 )                                                      R1T0169
 2000 FORMAT(1H1,/////, 27X,*EMPIRICALLY BASED COMPUTER PROGRAM* //     R1T0170
     1                  23X,*TO PREDICT THE AERODYNAMIC CHARACTERISTICS*R1T0171
     2 //               35X,*OF LARGE AIRCRAFT* (10(/)), 27X, 6A10 )    R1T0172
 2001 FORMAT(5X,*KPRINT(*,I2,*) = *,I2)                                 R1T0173
C                                                                       R1T0174
      END                                                               R1T0175
```

CC = 00044

```
      SUBROUTINE SETUP                                                  R1T0177
C                                                                       R1T0178
C     ZEROES OUT INPUT COMMON BLOCKS                                    R1T0179
C                                                                       R1T0180
      COMMON /BLKA01/ N(7)                                              R1T0181
      COMMON /BLKA02/ A02(433)                                          R1T0182
      COMMON /BLKA03/ A03(20)                                           R1T0183
      COMMON /BLKA04/ N3(3), A04(78)                                    R1T0184
      COMMON /BLKSUR/ N1(42), SUR(370)                                  R1T0185
      COMMON /BLKHLS/ N2, HLS(70)                                       R1T0186
      COMMON /BLKADJ/ IVAL(20),X(15),YM(135),YA(135),XCL(15),NXVAR,NXCL R1T0187
      COMMON /BLKG01/ G1(200)                                           R1T0188
C                                                                       R1T0189
      N2     = 0                                                        R1T0190
      NXVAR  = 1                                                        R1T0191
      NXCL   = 1                                                        R1T01911
C                                                                       R1T0192
      DO 10 I = 1, 20                                                   R1T0193
      A03(I) = 0.0                                                      R1T0194
      IVAL(I)= 0                                                        R1T0195
      IF( I.LE.3 ) N3(I) = 0                                            R1T0196
      IF ( I.LE.7 ) N(I) = 0                                            R1T0197
      IF ( I.LE.15) X(I) = 0.0                                          R1T0198
      IF( I.LE.15) XCL(I) = 0.0                                         R1T01991
   10 CONTINUE                                                          R1T0199
C                                                                       R1T0200
      DO 20 I = 1, 433                                                  R1T0201
      A02(I) = 0.0                                                      R1T0202
      IF( I.LE.41 ) N1(I) = 0                                           R1T0203
      IF( I.LE.50 ) HLS(I) = 0.0                                        R1T0204
      IF( I.LE.78 ) A04(I) = 0.0                                        R1T0205
      IF( I.LE.140) SUR(I) = 0.0                                        R1T0206
      IF( I.LE.200 ) G1(I) = 0.0                                        R1T0207
      IF( I.GT.135) GO TO 20                                            R1T0208
      YM(I)  = 1.0                                                      R1T0209
      YA(I)  = 0.0                                                      R1T0210
   20 CONTINUE                                                          R1T0211
C                                                                       R1T0212
      DO 30 I = 1, 20                                                   R1T0213
      I1     = I + 44                                                   R1T0214
      I2     = I + 204                                                  R1T0215
      I3     = I + 379                                                  R1T0216
      A02(I1) = 1.0                                                     R1T0217
      A02(I2) = 1.0                                                     R1T0218
      IF( I.LE.10 ) A02(I3) = 1.0                                       R1T0219
   30 CONTINUE                                                          R1T0220
C                                                                       R1T0221
      RETURN                                                            R1T0222
      END                                                               R1T0223

                                                            CC = 00049
```

```
      OVERLAY(1,1)
      PROGRAM INPT
C
C     PROBLEM DATA INPUT
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
      COMMON /BLKA02/ SREF, AR, TAPR, SWPLE,
     1                BLEN(10), BWID(10), BHGT(10), BAWET(10), BC(10),
     2                BNJ(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),
     3                BASE(10),  ELEN(10), EWID(10), EHGT(10),
     4                EAWET(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),
     5                ELBT(10), EQF(10), END(10), CBAR(10), TW,
     5                XLEW, YWW, YB, CR, BO2, BFUS, FMISC, AB, AFTAW,
     6                CAM(10), TOC(10), AWET(10), SWMT, SPLAN, CCNCL,
     7                TWIST, ETWIST, WINC, XLE(11), CRW(11), YW(11),
     8                XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC,
     9                SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),
     1                SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),
     2                SCR(10), HTLE, HTY, HTZ, HTINC
      COMMON /BLKA03/ ROUGHK,CLE(3),CCR(3),YC(3),XCG,ZCG,CMAC,IREF,A1(6)
      COMMON /BLKA04/ IHLS, NF, NS, FLAP, SLAT, DF(5), CFOC(5),
     1                DCLOF(5), DCLMF(5), DCDF(5), DCMOF(5),
     2                DS(5), CSOC(5), DCLOS(5), DCLMS(5), DCDS(5),
     3                DCMOS(5), BF1I, BF1O, CF1(3), BF2I, BF2O, CF2(3),
     4                BS1I, BS1O, CS1, BS2I, BS2O, CS2
      COMMON /BLKA05/ XMU,ZMU,XML,ZML,ZPTE,ZTHICK,
     1                RLE, OTE, DY, XMT, DYC,
     2                C1MAX, CLMCR(10), XMCR(10),
     3                XT(33), YT(33),XYC(33), NMCR, NXSET
C
      COMMON /BLKSUR/ NSURV, NCLAS, IT(20), ITRM(20), FMSURV(20),
     1                ALT(20), DHSV(20), SWPV(20), CLLO(20), CLHI(20),
     2                TRB(5,10), TRN(5,10), TRU(5,10), TRL(5,10),
     3                TRS(5,10)
      COMMON /BLKHLS/ NHLSV, DFI(3,5), CPF(5), DSI(5), CPS(5),
     1                DELCD(5), H(5), DFI2(3,5), CPF2(5), DSI2(5),
     2                CPS2(5)
C
      COMMON /BLKADJ/ IVAL(20),X(15),YM(15,9),YA(15,9),XCL(15),
     1                NXVAR, NXCL
C
      DIMENSION W(10), TTT(22), TS1(10), TS3(10)
C
      DATA  W   / 4HPLAI, 4HS.S., 4HD.S., 4HT.S., 4HINPU, 4HL.E.,
     1            4HSLAT, 4HKRUE, 4HINPU, 4HADJU/
      DATA  TTT / 3H63-,3H64-,3H65-,3H66-,3H63A,3H64A,3H65A,3HSUP,3HBIC,
     1            3H-62,3H-63,3H-64,3H-65,3H-66,3H-33,3H-34,3H-35,3H-93,
     2            3H-94,3H-95,3HINP,3H2IN /
      DATA  TS1 / 10*0.0 /, TS3 / 10* 0.0 /
C
C     CONFIGURATION DEFINITION
C
  100 READ(5,1001) NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS, IHLS,
     1            IREF
      WRITE(6,2000) NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
```

```
      READ(5,1000) SREF, CMAC, XCG, ZCG, ROUGHK, FMISC
      WRITE(6,2001) SREF, CMAC, XCG, ZCG
C
      IF( NBODYS.EQ.0 ) GO TO 120
      DO 110 I = 1, NBODYS
      READ(5,1000) BLEN(I), BWID(I), BHGT(I), BAWET(I), BQ(I), BNO(I)
      IF( I.EQ.1 ) READ(5,1008) BAMX(I), BABS(I), BLNS(I), BLBT(I),
     1                  BASE(I), BFUS, AB
      IF( I.NE.1 ) READ(5,1000) BAMX(I), BABS(I), BLNS(I), BLBT(I),
     1                          BASE(I)
C
      IF( BQ(I).EQ.0.0 ) BQ(I) = 1.0
      IF( BNO(I).EQ.0.0) BNO(I) =1.0
      IF( I.EQ.1 ) WRITE(6,2002)
      WRITE(6,2003) I,BLEN(I),BWID(I),BHGT(I),BAWET(I),BAMX(I),BABS(I),
     1              BLNS(I),BLBT(I),BASE(I),BQ(I), BNO(I)
C
  110 CONTINUE
C
  120 IF( NNACS.EQ.0 ) GO TO 140
      DO 130 I = 1, NNACS
      READ(5,1000) ELEN(I), EWID(I), EHGT(I), EAWET(I), EQF(I), ENO(I)
      READ(5,1000) EAMX(I), EIN(I), EXIT(I), ELNS(I), ELBT(I)
C
      IF( EQF(I).EQ.0.0 ) EQF(I) = 1.0
      IF( ENO(I).EQ.0.0 ) ENO(I) = 1.0
      IF( I.EQ.1 ) WRITE(6,2004)
      WRITE(6,2003) I,ELEN(I),EWID(I),EHGT(I),EAWET(I),EAMX(I),EIN(I),
     1                EXIT(I),ELNS(I),ELBT(I),EQF(I), ENO(I)
C
  130 CONTINUE
C
  140 IF( NPNLS.GT.1 ) GO TO 145
      READ(5,1000) AR, TAPR, SWPLE, SPLAN, TWIST, WINC
      WRITE(6,2006) AR, TAPR, SWPLE, SPLAN, TWIST, WINC
      READ(5,1004) TW1,TW2,TW3,CAM(1),TOC(1),XLEW,YWW, YB
      READ(5,1000) SWMT, CBAR(1), AWET(1), CONCL
C
      WRITE(6,2008) TW1,TW2,TW3, CAM(1), TOC(1), XLEW, YWW, YB,
     1              SWMT, CBAR(1), AWET(1)
      GO TO 160
C
  145 READ(5,1000) SPLAN, TAPR, SWPLE
      READ(5,1004) TW1,TW2,TW3, TWIST, WINC, SWMT, CONCL
      WRITE(6,2009) TW1,TW2,TW3, TWIST, WINC, SWMT
      IF( ISWP.GT.0 ) WRITE(6,2010) SPLAN, TAPR, SWPLE
C
      DO 150 I = 1, NPNLS
      IF( I.EQ.1 ) WRITE(6,2011)
      READ(5,1008) CAM(I), TOC(I), XLE(I), YW(I), CRW(I),
     1             CBAR(I), AWET(I)
      WRITE(6,2003)      I, CAM(I), TOC(I), XLE(I), YW(I), CRW(I),
     1              CBAR(I), AWET(I)
C
  150 CONTINUE
      I      = 1 + NPNLS
```

```
      READ(5,1000) XLE(I), YW(I), CRW(I)
      WRITE(6,2012) XLE(I), YW(I), CRW(I)
C
  160 IF( TW1.EQ.TTT(8) ) READ(5,1000) XMU,ZMU,XML,ZML,ZPTE,ZTHICK
C
      IF( TW1.EQ.TTT(21).OR.TW1.EQ.TTT(22) ) READ(5,1000)
     1        RLE, OTE, DY, XMT, DYC, CLMAX
      IF( TW1.EQ.TTT(21) ) READ(5,1009) NXSET, ( XT(I), YT(I),
     1                  XYC(I), I = 1,NXSET )
      IF( TW1.EQ.TTT(22) ) READ(5,1009) NMCR, ( CLMCR(I),
     1                                XMCR(I), I = 1, NMCR  )
      IF( ISWP.EQ.0 ) GO TO 170
      READ(5,1008) XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC, AFTAW
C
  170 IF( NSURFS.LE.1 ) GO TO 185
      DO 180 I = 2, NSURFS
      J       = I - 1
      IF( J.EQ.1 ) WRITE(6,2013)
      READ(5,1004) TS1(J),TS2,TS3(J), SCAM(J), STOC(J), SMTSW(J),
     1             SHF(J)
      READ(5,1000) SWL(J), SWT(J), STAPR(J), SCR(J), SBAR(J), SAWET(J)
      WRITE(6,2007) TS1(J),TS2,TS3(J), SCAM(J),STOC(J),SMTSW(J),
     1              SHF(J),SWL(J),SWT(J),STAPR(J),SCR(J),SBAR(J),
     2              SAWET(J)
      IF( NHT.NE.0.AND.J.EQ.1 ) READ(5,1000) HTLE, HTY, HTZ, HTINC
C
  180 CONTINUE
  185 IF( NPNLS.LE.2 ) GO TO 190
C
      READ(5,1000) (CLE(I), YC(I), CCR(I), I =1,3)
C
  190 CONTINUE
C
C
C     HIGH LIFT SYSTEM GEOMETRY IS READ IN IF IHLS GE 1
C
  200 IF( IHLS.EQ.0 ) GO TO 205
      READ(5,1007) FLAP, SLAT, NF, NS
      READ(5,1000) BF1I, BF1O, CF1(1), BS1I, BS1O, CS1
      IF( IHLS.EQ.2 ) READ(5,1000) BF2I,BF2O,CF2(1),BS2I,BS2O,CS2
      IF( FLAP.EQ.W(3).OR.FLAP.EQ.W(4) ) READ(5,1000)  CF1(2), CF1(3),
     1                CF2(2), CF2(3)
      IF( FLAP.EQ.W(5)) READ(5,1000) (DF(I), CFCC(I), DCLOF(I),
     1                         DCLMF(I), DCDF(I), DCMOF(I), I=1,NF )
C
      IF( SLAT.EQ.W(5)) READ(5,1000) (DS(I), CSOC(I), DCLOS(I),
     1                         DCLMS(I), DCDS(I), DCMOS(I), I=1,NS )
C
C     PROBLEM CONTROL
C
  205 READ(5,1003) NSURV, NHLSV, NCLAS, (IT(I),I=1,20), (ITRM(I),I=1,20)
      IF( NSURV.EQ.0 ) GO TO 220
      WRITE(6,2014) NSURV
      DO 210 I =1, NSURV
      READ(5,1000) FMSURV(I), ALT(I), DHSV(I), SWPV(I), CLLO(I),CLHI(I)
      IF( ITRM(I).EQ.0 ) WRITE(6,2015) FMSURV(I), ALT(I), DHSV(I),
```

```
     1                SWPV(I), CLLO(I), CLHI(I)
      IF( ITRM(I).EQ.1 ) WRITE(6,2016) FMSURV(I), ALT(I), SWPV(I),
     1                   CLLO(I), CLHI(I)
  210 CONTINUE
C
C     B.L. TRANSITION LOCATIONS ARE READ IN FOR EACH COMPONENT
C
      J1     = 0
      DO 215 I = 1, NSURV
      IF( IT(I).EQ.0 ) GO TO 215
      J      = IT(I)
      IF( J.LE.J1 ) GO TO 215
      IF( NBODYS.GT.0 ) READ(5,1000) (TRB(J,K), K=1,NBODYS)
      IF( NNACS.GT.0 )  READ(5,1000) (TRN(J,K), K=1,NNACS)
      IF( NPNLS.GT.0 )  READ(5,1000) (TRU(J,K),TRL(J,K),K=1,NPNLS)
      IF( NSURFS.GT.0)  READ(5,1000) (TRS(J,K), K =2, NSURFS)
      J1     = J
  215 CONTINUE
C
C
  220 IF( NHLSV.EQ.0 ) GO TO 250
      DO 230 I = 1, NHLSV
      NDF    = 1
      IF( FLAP.EQ.W(3) ) NDF = 2
      IF( FLAP.EQ.W(4) ) NDF = 3
C
      READ(5,1000) (DFI(J,I), J=1,NDF), CPF(I), DSI(I), CPS(I),
     1             DELCD(I), H(I)
      IF( IHLS.EQ.2 ) READ(5,1000) (DFI2(J,I), J=1,NDF), CPF2(I),
     1                DSI2(I), CPS2(I)
  230 CONTINUE
C
C
  250 READ(5,1005) WORD, WRD
      IF( WORD.EQ.W(10)) GO TO 260
      GO TO 270
  260 READ(5,1006) (IVAL(I), I=1,20), NXVAR, NADJ, NXCL, NADJ2
      IF(NXVAR.EQ.0) GO TO 266
      READ(5,1000) ( X(I), I =1, NXVAR )
      DO 265 I = 1, NADJ
      READ(5,1000) (YM(J,I), YA(J,I), J= 1, NXVAR )
  265 CONTINUE
  266 IF( NXCL.EQ.0 ) GO TO 268
      READ(5,1000) ( XCL(I), I =1, NXCL)
      DO 267 I = 1, NADJ2
      READ(5,1000) (YM(J,NXVAR+I), YA(J,NXVAR+I), J=1,NXCL)
  267 CONTINUE
  268 CONTINUE
      READ(5,1005) WORD, WRD
C
  270 WRITE(6,2005) WORD,WRD
C
C
      A      = 1./57.2957796
      B      = 1./12.
      C      = 1./144.
```

```
C
      SWPLE  = SWPLE * A
      SWMT   = SWMT * A
      AFTSW  = AFTSW * A
      DO 300 I = 1, 10
      SMTSW(I) = SMTSW(I) * A
      SWL(I) = SWL(I) * A
      SWT(I) = SWT(I) * A
  300 CONTINUE
C
      XPIVOT = XPIVOT * B
      YPIVOT = YPIVOT * B
      XAPEX  = XAPEX * B
      AFTCB  = AFTCB * B
      HTLE   = HTLE * B
      HTY    = HTY * B
      HTZ    = HTZ * B
      CMAC   = CMAC * B
      XCG    = XCG * B
      ZCG    = ZCG * B
      XLEW   = XLEW * B
      YWW    = YWW  * B
      YB     = YB   * B
      CR     = CR   * B
      BO2    = BO2  * B
      DO 310 I = 1, 10
      BLEN(I) = BLEN(I) * B
      BWID(I) = BWID(I) * B
      BHGT(I) = BHGT(I) * B
      BLNS(I) = BLNS(I) * B
      BLBT(I) = BLBT(I) * B
      ELEN(I) = ELEN(I) * B
      EWID(I) = EWID(I) * B
      EHGT(I) = EHGT(I) * B
      ELNS(I) = ELNS(I) * B
      ELBT(I) = ELBT(I) * B
      CBAR(I) = CBAR(I) * B
      XLE(I)  = XLE(I)  * B
      CRW(I)  = CRW(I)  * B
      YW(I)   = YW(I)   * B
      SBAR(I) = SBAR(I) * B
      SCR(I)  = SCR(I)  * B
  310 CONTINUE
      XLE(11) = XLE(11) * B
      CRW(11) = CRW(11) * B
      YW(11)  = YW(11)  * B
      DO 315 I = 1,3
      CLE(I)  = CLE(I) * B
      CCR(I)  = CCR(I)  * B
      YC(I)   = YC(I)  * B
  315 CONTINUE
C
      DO 320 I = 1, 10
      BAMX(I) = BAMX(I) * C
      BABS(I) = BABS(I) * C
      BASE(I) = BASE(I) * C
```

```
      EAMX(I) = EAMX(I) * C
      EIN(I)  = EIN(I)  * C
      EXIT(I) = EXIT(I) * C
  320 CONTINUE
C
      DO 340 I = 1, 10
      DO 330 J = 1, 22
C
      IF( TW1.EQ.TTT(J) ) TW = J
      IF( TW3.EQ.TTT(J) ) TW = J
      IF( TS1(I).EQ.TTT(J) ) TS(I) = J
      IF( TS3(I).EQ.TTT(J) ) TS(I) = J
C
  330 CONTINUE
  340 CONTINUE
C
      XFLAP  = FLAP
      XSLAT  = SLAT
      FLAP   = 0.0
      SLAT   = 0.0
C
      DO 350 I = 1,5
      IF(XFLAP.EQ.W(I)) FLAP = I
      J      = I + 5
      IF(XSLAT.EQ.W(J)) SLAT = I
  350 CONTINUE
C
 1000 FORMAT(6F10.0)
 1001 FORMAT(12I5)
 1002 FORMAT(F10.0, A3, F3.0, F4.0, 4F10.0)
 1003 FORMAT(3I5, 5X, 40I1 )
 1004 FORMAT(A3,A1,A3,3X, 5F10.0)
 1005 FORMAT( A4,A10)
 1006 FORMAT( 20I1, 4I5 )
 1007 FORMAT( A4, 16X, A4, 16X, 4I5 )
 1008 FORMAT(6F10.0,F6.0)
 1009 FORMAT( 1I5/(6F10.0) )
C
 2000 FORMAT(1H1, *  PROBLEM INPUT PARAMETERS*//5X, *NBODYS =*,I2,
     1        5X,*NNACS =*,I2,5X,*NSURFS =*,I2,5X,*NHT =*,I2,5X,
     2        *NVT =*,I2,5X,*ISWP =*,I2,5X,*NPNLS =*,I2 / )
 2001 FORMAT(5X,*SREF =*,F9.3,* SQ.FT.*,5X,*CMAC =*,F9.3,* IN.*,
     1       5X,*FUS. STA. C.G. =*F9.3,* IN.*,5X,*ZCG =*F8.3,* IN.*// )
 2002 FORMAT(T3,*NO.*, T10,*BLEN*, T20,*BWID*,T30,*BHGT*, T40,*BAWET*,
     1        T50,*BAMX*,T60,*BABS*,T70,*BLNS*,T80,*BLBT*,T90,*BASE*,
     2        T100,*BQ*, T110,*BNO* / )
 2003 FORMAT(I5, 11F10.3 )
 2004 FORMAT(/T3,*NO.*, T10,*ELEN*, T20,*EWID*,T30,*EHGT*, T40,*EAWET*,
     1        T50,*EAMX*,T60,*EIN*, T70,*EXIT*,T80,*ELNS*,T90,*ELBT*,
     2        T100,*EQF*,T110,*ENO* / )
 2005 FORMAT(//5X, A4, A10, 10X, 5H*****, /1H1  )
 2006 FORMAT(//* ASPECT RATIO =*F7.3, 5X,*TAPER RATIO =*F6.4,
     1        5X,*L.E.SWEEP =*F7.3, 5X,*PLANFORM AREA =*F8.2,* SQ.FT.*/
     2        5X,*TWIST =*F7.3,* DEG.*, 5X,*INCIDENCE =*F7.3,* DEG.* )
 2007 FORMAT(2X,A3,A1,A3,    11F10.3)
 2008 FORMAT(5X, *AIRFOIL TYPE =*,2X,A3,A1,A3,5X,*CAMBER =*F7.4,5X,
```

```
     1           *THICKNESS =*F8.4 /5X, *WING LOCATION (X,Y) = *,        R1T0561
     2           2F12.4,*FUSLAGE INTERSECTION =*F9.4 /5X,                 R1T0562
     3           *MAX. THICKNESS SWP. =*F7.3,5X,*CBAR =*F8.4,5X,          R1T0563
     4           *AWET =*F8.3 // )                                        R1T0564
 2009 FORMAT(/ 5X,*AIRFOIL TYPE =*,2X,A3,A1,A3,5X,*TWIST =*F7.3,* DEG.*,R1T0565
     1       5X,*INCIDENCE =*F7.3,* DEG.*,5X,*MAX. THICKNESS SWP. =*,   R1T0566
     2       F7.3,* DEG.*   )                                            R1T0567
 2010 FORMAT(5X,*VARIABLE GEOM. WING PLANFORM AREA =*F9.3,* SQ.FT.*,    R1T0568
     1       5X,*TAPER RATIO =*F6.4,5X,*L.E.SWEEP =*F7.3,* DEG.* /  )    R1T0569
 2011 FORMAT(/5X,*WING PANEL GEOMETRY*/T3,*NO.*,T10,*CAMBER*, T20,      R1T0570
     1       *T/C*, T30,*XLE*, T40,*YW*, T50,*CRW*, T60,*CBAR*,          R1T0571
     2       T70,*AWET* / )                                              R1T0572
 2012 FORMAT(25X,3F10.3 )                                                R1T0573
 2013 FORMAT(/5X,*GEOMETRY FOR ADDITIONAL AIRFOIL SURFACES* /            R1T0574
     1       T2,*AIRFOIL *,     T15,*CAMBER*, T25,*T/C*, T35,*SMTSW*,    R1T0575
     2       T45,*SHF*, T55,*L.E.SWP*, T65,*T.E.SWP*, T75,*TAPER*,       R1T0576
     3       T85,*CR*, T95,*SBAR*, T105,*SAWET* / )                      R1T0577
 2014 FORMAT(//5X, I5,* DRAG POLARS TO BE GENERATED AT THE FOLLOWING CONR1T0578
     1DITIONS* /T5, *MACH NO.*, T20,*ALTITUDE*, T35,*TAIL SETTING*,      R1T0579
     2          T53,*L.E.SWEEP*, T70,*FROM CL*, T85,*TO CL* / )          R1T0580
 2015 FORMAT( F15.4, 5F15.3)                                             R1T0581
 2016 FORMAT( F15.4, F15.3,15H       TRIMMED,  3F15.3)                   R1T0582
C                                                                        R1T0583
C                                                                        R1T0584
      END                                                                R1T0585

                                                            CC = 00361
```

```
      OVERLAY(1,2)
      PROGRAM NINPT
C
C     READS INPUT WITH A NAMELIST FORMAT
C
      COMMON /BLKOV1/ IJ
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
      COMMON /BLKA02/ SREF, AR, TAPR, SWPLE,
     1                BLEN(10), BWID(10), BHGT(10), BAWET(10), BQ(10),
     2                BNO(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),
     3                BASE(10),  ELEN(10), EWID(10), EHGT(10),
     4                EAWET(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),
     5                ELBT(10), EQF(10), ENO(10), CBAR(10), TW,
     5                XLEW, YWW, YB, CR, BO2, BFUS, FMISC, AB, AFTAW,
     6                CAM(10), TOC(10), AWET(10), SWMT, SPLAN, CONCL,
     7                TWIST, ETWIST, WINC, XLE(11), CRW(11), YW(11),
     8                XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC,
     9                SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),
     1                SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),
     2                SCR(10), HTLE, HTY, HTZ, HTINC
      COMMON /BLKA03/ ROUGHK,CLE(3),CCR(3),YC(3),XCG,ZCG,CMAC,IREF,A1(6)
      COMMON /BLKA04/ IHLS, NF, NS, FLAP, SLAT, DF(5), CFOC(5),
     1                DCLOF(5), DCLMF(5), DCDF(5), DCMOF(5),
     2                DS(5), CSOC(5), DCLOS(5), DCLMS(5), DCDS(5),
     3                DCMOS(5), BF1I, BF1O, CF1(3), BF2I, BF2O, CF2(3),
     4                BS1I, BS1O, CS1, BS2I, BS2O, CS2
C
C
      COMMON /BLKSUR/ NSURV, NCLAS, IT(20), ITRM(20), FMSURV(20),
     1                ALT(20), DHSV(20), SWPV(20), CLLO(20), CLHI(20),
     2                TRB(5,10), TRN(5,10), TRU(5,10), TRL(5,10),
     3                TRS(5,10)
      COMMON /BLKHLS/ NHLSV, DFI(3,5), CPF(5), DSI(5), CPS(5),
     1                DELCD(5), H(5), DFI2(3,5), CPF2(5), DSI2(5),
     2                CPS2(5)
C
      COMMON /BLKADJ/ IVAL(20),X(15),YM(15,9),YA(15,9),XCL(15),
     1                NXVAR, NXCL
      COMMON /BLKTIL/ TITLE(6)
      COMMON /BLKPRT/ KPRINT(50)
C
      DIMENSION L(9)
C
      NAMELIST/BODYS/ NBODYS,NNACS,NSURFS,NHT,NVT,ISWP,NPNLS,CMAC,IREF,
     1                SREF,ROUGHK,XCG,ZCG,BFUS,AB,FMISC,BLEN,BWID,BHGT,
     2                BAWET,BQ,BNO,BAMX,BABS,BLNS,BLBT,BASE/
     3          NACEL/ NNACS,SREF,ELEN,EWID,EHGT,EAWET,EQF,ENO,
     4                EAMX,EIN,EXIT,ELNS,ELBT/
     5           WING/ NSURFS,ISWP,NPNLS, AR,TAPR,SWPLE,SPLAN,TWIST,
     6                WINC,TW,CAM,TOC,XLE,YW,CRW,CBAR,AWET,SWMT,CMAC,
     7                XPIVOT,YPIVOT,XAPEX,AFTSW,AFTCB,AFTOC,SREF,AFTAW,
     8                CLE,YC,CCR,XLEW,YWW,YB,CR,BO2,XCG,CONCL/
     9          SURFS/ NSURFS,NHT,NVT,SREF, SBAR,TS,SCAM,STOC,SAWET,
     0                SMTSW,SHF,SWL,SWT,STAPR,SCR,HTLE,HTY,HTZ,HTINC/
     4           SURV/ NSURV, NCLAS, NHLSV, IT,ITRM,FMSURV,ALT,DHSV,SWPV,
     A                CLLO,CLHI,TRB,TRN,TRU,TRL,TRS,DFI,CPF,DSI,CPS,
```

```
     B                    DELCO,H,DFI2,CPF2,DSI2,CPS2,KPRINT/
     C             STOL/  IHLS,NF,NS,FLAP,SLAT,DF,CFOC,DCLOF,DCLMF,DCOF,
     D                    DCMOF,DS,CSOC,DCLOS,DCLMS,DCDS,DCMCS,BF1I,
     E                    BF1O,CF1,BF2I,BF2O,CF2,BS1I,BS1O,CS1,BS2I,
     F                    BS2O,CS2,H,DFI2,CPF2,DSI2,CPS2/
     C           ADJUST/  IVAL, NXVAR, NXCL, X, XCL, YM, YA
C
      DO 100 I = 1,9
      L(I)   = 1
  100 CONTINUE
C
      CALL CONV
      IF( IJ.EQ.1 ) GO TO 200
      READ(5,1000) (TITLE(I), I =1,6)
      READ(5,1001) (L(I), I = 1,9)
C
  200 IF( L(1).EQ.1 ) READ(5,BODYS)
      IF( L(2).EQ.1 ) READ(5,NACEL)
      IF( L(3).EQ.1 ) READ(5,WING)
      IF( L(4).EQ.1 ) READ(5,SURFS)
      IF( L(5).EQ.1 ) READ(5,SURV)
      IF( L(6).EQ.1 ) READ(5,STOL)
      IF( L(7).EQ.1 ) READ(5,ADJUST)
C
      WRITE(6,2000)
C
      CALL CONV1
C
  350 CONTINUE
 1000 FORMAT( 6A10 )
 1001 FORMAT( 50I1 )
 2000 FORMAT(5X,*NAMELIST INPUT COMPLETED* )
C
      END
```

CC = 00090

```
      SUBROUTINE CONV
C
C     CONVERTS INPUT DATA TO FEET AND RADIANS UNITS
C
      COMMON /BLKA02/ A1(433)
      COMMON /BLKA03/ A2(20)
C
      DIMENSION B1(433), B2(20)
C
      DO 100 I = 1, 20
  100 B2(I)  = A2(I)
C
      DO 200 I = 1, 433
  200 B1(I)  = A1(I)
C
      RETURN
C
      ENTRY CONV1
C
      A      = 1./57.2957796
      B      = 1./12.
      C      = 1./144.
C
      IF( A1(4).NE.B1(4) ) A1(4) = A1(4) * A
      IF( A1(275).NE.B1(275) ) A1(275) = A1(275) * A
      IF( A1(317).NE.B1(317) ) A1(317) = A1(317) * A
      DO 300 I = 1, 10
      IF( A1(369+I).NE.B1(369+I)) A1(369+I) = A1(369+I)*A
      IF( A1(389+I).NE.B1(389+I)) A1(389+I) = A1(389+I)*A
      IF( A1(399+I).NE.B1(399+I)) A1(399+I) = A1(399+I)*A
  300 CONTINUE
C
      DO 400 I = 1, 10
      IF( A1(224+I).NE.B1(224+I)) A1(224+I) = A1(224+I) * B
      IF( A1(319+I).NE.B1(319+I)) A1(319+I) = A1(319+I) * B
      IF( A1(419+I).NE.B1(419+I)) A1(419+I) = A1(419+I) * B
      IF( A1(104+I).NE.B1(104+I)) A1(104+I) = A1(104+I) * C
      IF( A2(1+I).NE.B2(1+I) ) A2(1+I) = A2(1+I) * B
  400 CONTINUE
C
      DO 500 I = 1, 20
      IF( A1(84 +I).NE.B1(84 +I)) A1(84 +I) = A1(84 +I) * B
      IF( A1(184+I).NE.B1(184+I)) A1(184+I) = A1(184+I) * B
      IF( A1(64 +I).NE.B1(64 +I)) A1(64 +I) = A1(64 +I) * C
  500 CONTINUE
C
      DO 600 I = 1, 30
      IF( A1(4 + I).NE.B1(4 + I)) A1(4 + I) = A1(4 + I) * B
      IF( A1(114+I).NE.B1(114+I)) A1(114+I) = A1(114+I) * B
      IF( A1(280+I).NE.B1(280+I)) A1(280+I) = A1(280+I) * B
      IF( A1(154+I).NE.B1(154+I)) A1(154+I) = A1(154+I) * C
  600 CONTINUE
C
      IF( A1(314).NE.B1(314) ) A1(314) = A1(314) * B
      IF( A1(315).NE.B1(315) ) A1(315) = A1(315) * B
      IF( A1(316).NE.B1(316) ) A1(316) = A1(316) * B
```

```
      IF( A1(318).NE.B1(318) ) A1(318) = A1(318) * B
      IF( A1(430).NE.B1(430) ) A1(430) = A1(430) * B
      IF( A1(431).NE.B1(431) ) A1(431) = A1(431) * B
      IF( A1(432).NE.B1(432) ) A1(432) = A1(432) * B
      IF( A1(236).NE.B1(236) ) A1(236) = A1(236) * B
      IF( A1(237).NE.B1(237) ) A1(237) = A1(237) * B
      IF( A1(238).NE.B1(238) ) A1(238) = A1(238) * B
      IF( A1(239).NE.B1(239) ) A1(239) = A1(239) * B
      IF( A1(240).NE.B1(240) ) A1(240) = A1(240) * B
      IF( A1(241).NE.B1(241) ) A1(241) = A1(241) * B
      IF( A1(291).NE.B1(291) ) A1(291) = A1(291) * B
      IF( A1(302).NE.B1(302) ) A1(302) = A1(302) * B
      IF( A1(313).NE.B1(313) ) A1(313) = A1(313) * B
      IF( A2(12).NE.B2(12) ) A2(12) = A2(12) * B
      IF( A2(13).NE.B2(13) ) A2(13) = A2(13) * B
C
      RETURN
      END
```

CC = 00074

```
      OVERLAY(2,0)
      PROGRAM GEOM
C
C     COMPUTES GEOMETRIC PARAMETERS FOR AERO CALCULATIONS
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
      COMMON /BLKA02/ SREF, AR, TAPR, SWPLE,
     1                BLEN(10), BWID(10), BHGT(10), BAWET(10), BC(10),
     2                BNO(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),
     3                BASE(10),  ELEN(10), EWID(10), EHGT(10),
     4                EAWET(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),
     5                ELBT(10), EQF(10), ENO(10), CBAR(10), TW,
     5                XLEW, YWW, YBOD, CR, BO2, A2(4),
     6                CAM(10), TOC(10), AWET(10), SWMT, SPLAN, CONCL,
     7                TWIST, ETWIST, WINC, XLE(11), CRW(11), YW(11),
     8                XPIVOT, YPIVOT, XAPEX, AFTSK, AFTCB, AFTOC,
     9                SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),
     1                SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),
     2                SCR(10), HTLE, HTY, HTZ, HTINC
      COMMON /BLKA03/ ROUGHK, CLE(3),CCR(3),YC(3), XCG,ZCG,CMAC,A1(7)
C
      COMMON /BLKG01/ FRBOD(10), ARS(10), SSEX(10), FRNAC(10), SWP,
     1                SWPQC, SWPMC, SWPTE, DOB, TOCW, CLO, SEXW, ESWMC,
     2                SXX(10), XLESWX(10), XMCSWX(10), ARW, TR, WPLAN,
     3                CB, YB, XB,
     4                CRX, CBX, CTX, YIX, YOX, SIX, SCX, ARI, ARXR,
     5                CBXP, AROP, SOXP, SWPLEI, SWPLEO, SWPMCI, SWPMCO,
     6                ESWQC, ESWLE, ESWTE, CROB, DXOB, XHT, XCRTE,
     7                XH, OMEGA, FOC, TWST, DINC, G1(5), DXQC, G3(81)
C
      COMMON /BLKPRT/ KPRINT(50)
C
      DIMENSION FOCX(22)
      DIMENSION G2(200)
      EQUIVALENCE (G2(1), FRBOD(1))
      DATA FOCX / 4*0.05515,3*0.06651,15*0.079 /
C
      TWST    = TWIST
      DINC    = WINC
C
      DO 100 I = 1, NBODYS
      BDIA2   = BWID(I) * BHGT(I)
      FRBOD(I) = 0.0
      IF( BAMX(I).EQ.0.0 ) BAMX(I) = 0.7854 * BDIA2
      IF( BAWET(I).EQ.0.0 ) BAWET(I) = (2.8*BLNS(I) +2.5*BLBT(I)
     1       *(1.+SQRT(BABS(I)/BAMX(I))) + (BLEN(I) -BLNS(I) -BLBT(I))
     2        *4. ) * SQRT(.7854 * BAMX(I)) * BNO(I)
      IF( BDIA2.GT.0.0 ) FRBOD(I) = BLEN(I)/SQRT(BDIA2)
  100 CONTINUE
C
      DO 200 I = 1, NNACS
      EDIA2   = EWID(I) * EHGT(I)
      FRNAC(I) = 0.0
      IF( EAMX(I).EQ.0.0 ) EAMX(I) = 0.7854 * EDIA2
      IF( EAWET(I).EQ.0.0 ) EAWET(I) = (2.5*ELNS(I) * (1. +
     1        SQRT(EIN(I)/EAMX(I)))    + 2.5 *ELBT(I)
```

```
     1        *(1.+SQRT(EXIT(I)/EAMX(I))) + (ELEN(I) -ELNS(I) -ELBT(I))  R1T0809
     2        *4.) * SQRT(.7854 * BAMX(I)) * ENO(I)                      R1T0810
      IF( EDIA2.GT.0.0 ) FRNAC(I) = ELEN(I)/SQRT(EDIA2)                R1T0811
  200 CONTINUE                                                         R1T0812
C                                                                      R1T0813
  210 NP      = NPNLS + 1                                              R1T0814
      IF( NPNLS.NE.1 ) GO TO 220                                       R1T0815
      TSWP    = TAN(SWPLE)                                             R1T0816
      CO      = 4./AR  *(1.-TAPR)/(1.+TAPR)                            R1T0817
C                                                                      R1T0818
C     GEOMETRY FOR SIMPLE TRAPEZOIDAL WINGS IS CALCULATED HERE         R1T0819
C                                                                      R1T0820
      SWP     = SWPLE                                                  R1T0821
      SWPQC   = ATAN(TSWP - CO * 0.25)                                 R1T0822
      SWPMC   = ATAN(TSWP - CO * 0.50)                                 R1T0823
      SWPTE   = ATAN(TSWP - CO )                                       R1T0824
      BO2     = 0.5 * SQRT(SPLAN * AR)                                 R1T0825
      YW(1)   = YBOD                                                   R1T0826
      XLE(1)  = XLEW + (YBOD - YWW) * TSWP                             R1T0827
      DOB     = YW(1)/BO2                                              R1T0828
C                                                                      R1T0829
      CR      = SPLAN/(BO2*(1.+TAPR))                                  R1T0830
      XCRTE   = XLEW -YWW * TSWP + CR                                  R1T0831
      CRX     = CR * (1. - DOB*(1.-TAPR) )                             R1T0832
      CTX     = CR * TAPR                                              R1T0833
      IF( CBAR(1).EQ.0.0 ) CBAR(1) = (CRX + CTX**2/(CRX+CTX))*0.66667  R1T0834
      YIX     = BO2 * (1.-DOB)                                         R1T0835
      CB      = 0.66667 * CR * (1. + TAPR*TAPR/(1. + TAPR))            R1T0836
      YB      = BO2/3. * (1. + 2.*TAPR)/(1.+ TAPR)                     R1T0837
      XB      = YB * TSWP + XLE(1) - YW(1)*TSWP                        R1T0838
C                                                                      R1T0839
      ESWMC   = SWPMC                                                  R1T0840
      ESWQC   = SWPQC                                                  R1T0841
      ESWLE   = SWPLE                                                  R1T0842
      ESWTE   = SWPTE                                                  R1T0843
      TOCW    = TOC(1)                                                 R1T0844
      CLD     = CAM(1)                                                 R1T0845
      TR      = TAPR                                                   R1T0846
      SEXW    = SPLAN/(1.+TR) * (2.-(1.-TR)*(YW(1)/BO2 +1.))           R1T0847
     1                * (1.-YW(1)/BO2)                                 R1T0848
      SXX(1) = SEXW                                                    R1T0849
      IF( AWET(1).EQ.0.0 ) AWET(1) = SEXW * (2. +.1843*TOCW            R1T0850
     1                   +1.5268*TOCW**2 -.8395*TOCW**3)               R1T0851
      XLESWX(1) = SWPLE                                                R1T0852
      XMCSWX(1) = SWPMC                                                R1T0853
      ARW     = AR                                                     R1T0854
      WPLAN   = SPLAN                                                  R1T0855
      SIX     = SEXW                                                   R1T0856
      ARXR    = 4.* YIX**2/SIX                                         R1T0857
      SWPLEI  = SWP                                                    R1T0858
      SWPMCI  = SWPMC                                                  R1T0859
C                                                                      R1T0860
      YW(2)   = BO2                                                    R1T0861
      XLE(2)  = XLE(1) + (YW(2)-YW(1)) * TSWP                          R1T0862
      CRW(1)  = CRX                                                    R1T0863
      CRW(2)  = CTX                                                    R1T0864
```

```
      CROB    = 0.5 * CR/BO2
      DXOB    = CROB - 0.75 * TAN(SWPQC)
C
  220 IF( NPNLS.GT.2 ) GO TO 240
      DO 230 I = 1, NP
      CLE(I) = XLE(I)
      CCR(I) = CRW(I)
      YC(I)  = YW(I)
  230 CONTINUE
C
  240 IF(NPNLS.LE.1) GO TO 310
C     GEOMETRY FOR MULTIPLE PANEL WINGS IS CALCULATED HERE
C
      WPLAN   = SPLAN
      BO2     = YW(NP)
      TR      = TAPR
      SWP     = SWPLE
      ETA     = YW(1)/YW(2)
      CRCL    = (CRW(1) - ETA * CRW(2)) /(1.-ETA)
      CROB    = 0.5 * CRCL/BO2
      DOB     = YW(1)/BO2
      SFXW    = 0.0
      TOCW    = 0.0
      CLD     = 0.0
      COSMC   = 0.0
      COSQC   = 0.C
      TANLE   = 0.0
      TANTE   = 0.0
      SUMCB   = 0.0
      SUMYB   = 0.0
      SUMXB   = 0.0
C
      DO 300 I =1, NPNLS
      DYW     = YW(I+1) - YW(I)
      SX      = (CRW(I+1) + CRW(I)) * DYW
      DXLE    = -XLE(I) + XLE(I+1)
      XLESW   = ATAN(DXLE/DYW)
      DXTE    = DXLE  + (CRW(I+1) - CRW(I))
      IF( I.EQ.1 ) XCRTE = XLE(1) + YW(1)*TAN(XLESW) + CRCL
      DX      = DXLE + 0.5 * (CRW(I+1) - CRW(I))
      DXQ     = DXLE + 0.25 * (CRW(I+1) - CRW(I))
      XTESW   = ATAN(DXTE/DYW)
      XMCSW   = ATAN(DX/DYW)
      XQCSW   = ATAN(DXQ/DYW)
C
      CR      = CRW(I)
      CT      = CRW(I+1)
      T       = CT/CR
C
      SXX(I) = SX
      XLESWX(I) = XLESW
      XMCSWX(I) = XMCSW
C
      CB      = 0.66667 * CR * (1. + T*T/(1.+T))
      IF( CBAR(I).EQ.0.0 ) CBAR(I) = CB
      YB      = 0.33333 * DYW * (1.+2.*T)/(1.+T) + YW(I)
```

```
      XB     = XLE(I) - XLE(1) + (YB - YW(I))*TAN(XLESW)
C
      DYX1   = 0.75*BO2 - YW(I)
      DYX2   = 0.75*BO2 - YW(I+1)
      IF( DYX1.GE.0.0.AND.DYX2.LT.0.0 ) X1 = XLE(I) + CRW(I)*0.25
     1               + DYX1 * TAN(XQCSW)
C
C
      IF( AWET(I).EQ.0.0 ) AWET(I) = SX * (2. +.1843*TOC(I)
     1                 +1.5268*TOC(I)**2 -.8395*TOC(I)**3)
      TOCW   = TOCW + SX * TOC(I)**2
      COSMC  = COSMC + COS(XMCSW) * SX
      COSQC  = COSQC + COS(XQCSW) * SX
      TANLE  = TANLE + TAN(XLESW) * SX
      TANTE  = TANTE + TAN(XTESW) * SX
      CLD    = CLD + CAM(I)**2 * SX
      SEXW   = SEXW + SX
C
      SUMCB  = SUMCB + CB*SX
      SUMYB  = SUMYB + YB*SX
      SUMXB  = SUMXB + XB*SX
C
  300 CONTINUE
C
      TOCW   = SQRT(TOCW/SEXW)
      CLD    = SQRT(CLD/SEXW)
      ESWMC  = ACOS(COSMC/SEXW)
      ESWQC  = ACOS(COSQC/SEXW)
      ESWLE  = ATAN(TANLE/SEXW)
      ESWTE  = ATAN(TANTE/SEXW)
C
      CB     = SUMCB/SEXW
      YB     = SUMYB/SEXW
      XB     = SUMXB/SEXW + XLE(1)
      X2     = XLE(1) - YW(1)*TAN(XLESWX(1)) + 0.75 * CRCL
      DXOB   = (X2 - X1)/BO2
C
      IF( WPLAN.EQ.0.0 ) WPLAN = SEXW + (CRCL + CRW(1)) * YW(1)
      ARW    = 4. * BO2**2/WPLAN
      TRK    = (TAN(ESWLE) - TAN(ESWTE) ) *0.25 * ARW
      IF( ISWP.EQ.0 ) TR = (1.-TRK)/(1.+TRK)
      TRLIM  = CRW(NP)/CRCL
      IF( TR.LT.TRLIM ) TR = TRLIM
C
  310 IF( NSURFS.LE.1 ) GO TO 410
      XNVT   = NVT
      DO 400 J = 2, NSURFS
      I      = J-1
C
      SSEX(I) = SAWET(I)*0.5
      IF(SCR(I).LE.0.0) GO TO 400
      ARS(I)  = SSEX(I)/SCR(I)**2
C
      IF(SWL(I).EQ.SWT(I)) GO TO 400
      ARS(I) = 4.*(1.-STAPR(I))/(1.+STAPR(I))/(TAN(SWL(I))-TAN(SWT(I)))
      SSEX(I) = (SCR(I) * (1.+STAPR(I)))**2 * ARS(I) * 0.25
```

```
      IF(NHT.EQ.0.AND.NVT.GT.0) SSEX(1) = 0.5 * XNVT * SSEX(1)
      IF(NHT.EQ.1.AND.NVT.GT.0) SSEX(2) = 0.5 * XNVT * SSEX(2)
C
      IF( SAWET(I).EQ.0.0 ) SAWET(I) = SSEX(I) * (2. +.1843*STOC(I)
     1                  +1.5268*STOC(I)**2 -.8395*STOC(I)**3)
      IF( SBAR(I).EQ.0.0 ) SBAR(I) = 0.66667 * SCR(I) *
     1                        (1.+STAPR(I)**2/(1.+STAPR(I)))
C
  400 CONTINUE
C
  410 IF( XCG.EQ.0.0 ) XCG = XB + 0.25 * CB
      IF( CMAC.EQ.0.0 ) CMAC = CB
      NI     = TW
      FOC    = CLD * FOCX(NI)
      DXQC   = XLE(1+NPNLS)+CRW  +NPNLS)*0.25  -XLE(1)-CRW(1)*0.25
C
C     GEOMETRY FOR THE HORIZONTAL TAIL IS CALCULATED
C
      XH     = 0.0
      OMEGA  = 0.0
C
      IF( NHT.EQ.0 ) GO TO 415
      BHT    = SQRT(ABS(1) * SSEX(1))
      XHT    = BHT/6. * (1. + 2.*STAPR(1))/(1.+ STAPR(1))*TAN(SWL(1))
     1                + SBAR(1) * 0.25 + HTLE
C
      XH     = SQRT((HTZ-ZCG)**2 + (XHT-XCG)**2)
      OMEGA  = ATAN((HTZ-ZCG)/(XHT-XCG))
C
  415 IF( NPNLS.LE.1 ) GO TO 420
C
C     CRANKED WING GEOMETRY IS CALCULATED
C
      CRX    = CCR(1)
      CBX    = CCR(2)
      CTX    = CCR(3)
      YIX    = YC(2) - YC(1)
      YOX    = YC(3) - YC(2)
C
      SIX    = (CRX + CBX) * YIX
      SOX    = (CBX + CTX) * YOX
      ARI    = 4. * YIX**2 / SIX
      ARXR   = 4. * (YIX+YOX)**2 /(SIX+SOX)
C
      DYP    = YIX * 0.5
      CBXP   = CTX + (YOX + DYP) * (CBX-CTX)/YOX
      SOXP   = (CBXP + CTX) * (YOX + DYP)
      AROP   = 4. * (YOX + DYP)**2 /SOXP
C
      SWPLEI = ATAN((CLE(2)-CLE(1))/YIX)
      SWPLEO = ATAN((CLE(3)-CLE(2))/YOX)
      SWPMCI = ATAN((CLE(2)-CLE(1)+(CBX-CRX)*0.5)/YIX)
      SWPMCO = ATAN((CLE(3)-CLE(2)+(CTX-CBX)*0.5)/YOX)
C
  420 IF( KPRINT(12).EQ.0 ) GO TO 421
      WRITE(6,1000) (G2(I), I = 1, 111)
```

```
C
  421 WRITE(6,2000)
      IF( NBODYS.GT.0 ) WRITE(6,2001) (I,BLEN(I),BAWET(I),FRBOD(I),
     1              BQ(I), I = 1, NBODYS)
C
      IF( NNACS.GT.0 )  WRITE(6,2002) (I,ELEN(I),EAWET(I),FRNAC(I),
     1              EQF(I), I = 1,NNACS)
C
      IF( NSURFS.EQ.0 ) GO TO 450
      X       = SWMT * 57.2958
      DO 425 I = 1, NPNLS
      IF( I.NE.NPNLS ) WRITE(6,2003) I, CBAR(I), AWET(I), TOC(I)
  425 IF( I.EQ.NPNLS ) WRITE(6,2003) I, CBAR(I), AWET(I), TOC(I), X
C
      NS      = NSURFS - 1
      IF( NS.LE.0 ) GO TO 450
      DO 430 I = 1, NS
      J       = I +1
      X       = SMTSW(I) * 57.2958
      WRITE(6,2004) J, SBAR(I), SAWET(I), STOC(I), SHF(I), X
  430 CONTINUE
  450 CONTINUE
 1000 FORMAT(5X, *GEOMETRY DATA DUMP,  COMMON BLKG01*/(1X,10F12.4))
 2000 FORMAT(1H1, 5X, *CONFIGURATION SUMMARY* ///
     1         31X, *LENGTH*, 7X, *WETTED AREA*, 5X, *FR OR T/C*,
     2         4X, *INTERFERENCE*, 4X, *MAX. T/C SWEEP* /
     3         30X, * (FT.)*, 8X, * (SQ. FT.) *, 20X,
     4         *FACTOR*, 8X, * (DEG.) * / )
 2001 FORMAT(5X, *BODY NUMBER* I2, 5X, 4F15.5)
 2002 FORMAT(5X, *NACELLE NO.* I2, 5X, 4F15.5)
 2003 FORMAT(5X, *WING PANEL NO.*I2, 2X, 3F15.5, 15X, F15.5)
 2004 FORMAT(5X, *AIRFOIL SURFACE NO.*I2, F12.5, 4F15.5)
C
      END
```

CC = 00314

```
      OVERLAY(3,0)
      PROGRAM SURVEY
C
C     AERODYNAMIC SURVEY
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON/BLKSUR/ NSURV, NCLAS, IT(20), ITRM(20), FMSURV(20),
     1               ALT(20), DHSV(20), SWPV(20), CLLO(20), CLHI(20),
     2               TRBNWS(250)
C
      COMMON /BLKHLS/ NHLSV, HLS(70)
C
      COMMON /BLKTIL/ TITLE(6)
      COMMON /BLKOV3/ ITYP, XMACH, TOC, CLD, SWPMC, H, FM, ITL, JPASS,
     1               ROT, OTE, ITRIM
C
      COMMON /BLKC01/ CL, CD, CM, ALPHA, CDM, CDL, CDR, CDRO, CLT, CDT,
     1               DM, FK, DELCL, CMO, DCMCL, XACWB, CLA, ALO,
     2               TCD(5), CDFUS(5), CDBOD(5), CDNAC(4), CDWING(4),
     3               CDHT(4), CDVT(4), CDSURF(4),
     4               C1(9), CLPB, CLDB, CLMAX, C2(9), CDC, C3(10),
     5               C4(7), CDMISC, DEDA, CDAFT, C5(5)
      COMMON /BLKA02/ A1(433)
      COMMON /BLKA03/ A2(13), IREF, A3(6)
      COMMON /BLKVGM/ KPASS, SWP, VGM(27)
      COMMON /BLKC03/ C03(22)
C
      COMMON /BLKG01/ G1(200)
C
      COMMON /BLKB02/ AA(22), BB(22), CC(22), DD(22), XT(22)
C
      DIMENSION  XOTE(20), YOTE(20)
      DATA XOTE / 34.6,38.4,46.4,60.2,57.5,59.5,66.5,30.0,95.0,50.0,
     1           63.0,82.8,113.0,153.0,63.0,82.8,113.0,63.0,82.8,113./,
     2     YOTE / 4*14.8,3*14.05,40.0,0.0,11*13.8 /
C
      IF( ISWP.EQ.0 ) GO TO 100
      KPASS  = 1
      SWP    = A1(4)
      CALL OVERLAY(4HOVLY,3,1)
      KPASS  = 2
      SWP    = A1(317)
      CALL OVERLAY(4HOVLY,3,1)
  100 KPASS  = 0
C
      TOC    = G1(46)
      CLD    = G1(47)
      ITYP   = A1(235)
      WINC   = A1(280)
      IF( NSURV.EQ.0 ) GO TO 510
C
      SWP2   = SWPV(1)
C
      JPASS  = 0
      DO 500 L = 1, NSURV
```

```
      FM       = FMSURV(L)
      H        = ALT(L)
      DH       = DHSV(L)
      SWEEP    = SWPV(L)
      CLLOW    = CLLO(L)
      ITL      = IT(L)
      ITRIM    = ITRM(L)
C
      XNCL     = NCLAS - 1
      DCL      = 0.0
      IF( XNCL.GT.0.0 ) DCL = (CLHI(L) - CLLO(L))/XNCL
C
      SWP      = SWEEP/57.2957796
      IF( ISWP.GT.0.AND.SWEEP.NE.SWP2 ) JPASS = 0
      SWP2     = SWEEP
C
      IF( ISWP.GT.0.AND.JPASS.EQ.0 ) CALL OVERLAY(4HOVLY,3,1)
      EO2      = 0.25 * SQRT(G1(80)*G1(82))
      DO 200 IP = 1, NPNLS
      SWPMW    = G1(69+IP)
      IF( ISWP.GT.0.AND.IP.EQ.NPNLS ) GO TO 190
      IF(A1(302+IP).LE.EO2.AND.A1(303+IP).GT.EO2)  GO TO 210
      GO TO 200
  190 IF( A1(302+NPNLS).LE.EO2 )  SWPMW = G1(69+NPNLS)
  200 CONTINUE
C
  210 IF( SWPMW.GT.G1(49) )  SWPMW = G1(49)
      SWPMC    = SWPMW * 57.2958
      TOG      = G1(46)
      CLD      = G1(47)
      ITYP     = A1(235)
      XMACH    = 0.6
C
      IF( JPASS.EQ.0 ) CALL OVERLAY(4HOVLY,3,2)
C
      IF( JPASS.EQ.2  ) JPASS = 1
C
      DO 300 M = 1, NCLAS
      XM1      = M-1
      CL       = CLLOW + DCL * XM1
C
      CALL OVERLAY(4HOVLY,3,3,6HRECALL)
C
      SWEEP    = G1(41) * 57.29578
      IF( M.NE.1 ) GO TO 220
      WRITE(6,1005) (TITLE(K), K=1,6)
      IF( H.GE.0.0.AND.ITRIM.EQ.0 ) WRITE(6,1000) FM,H,SWEEP,DH
      RN       = -H * 1.0E6
      IF( H.LT.0.0.AND.ITRIM.EQ.0 ) WRITE(6,1006) FM,RN,SWEEP,DH
      IF( H.GE.0.0.AND.ITRIM.NE.0 ) WRITE(6,1008) FM, H, SWEEP
      IF( H.LT.0.0.AND.ITRIM.NE.0 ) WRITE(6,1009) FM,RN, SWEEP
  220 ANGLE    = ALPHA
      IF( IREF.EQ.1 ) ANGLE = ALPHA - WINC
C
      CDRA     = CDR + CDAFT
      IF( ITRIM.EQ.0 ) WRITE(6,1001) CLT, CDT, CM, ANGLE, CDL, CDRA,
```

```
     1                                  CL, CD
      IF( ITRIM.EQ.1 ) WRITE(6,1001) CLT, CDT, DH, ANGLE, CDL, CDRA,
     1                                  CL, CD
C
  300 CONTINUE
C
      ALOR    = ALO
      IF( IREF.EQ.1 ) ALOR   = ALO - WINC
      WRITE(6,1002) CLA, ALOR, CLDB, FK, DELCL, CLMAX, CMO, DCMCL
C
      WRITE(6,1003)(TCD(J), CDFUS(J), CDBOD(J), CDNAC(J), CDWING(J),
     1              CDHT(J), CDVT(J), CDSURF(J),   J = 1, 4)
C
      WRITE(6,1004) TCD(5), CDFUS(5), CDBOD(5), CDC, CDRO, CDMISC
      WRITE(6,1007) CDM
C
C
  500 CONTINUE
C
  510 IF( NHLSV.EQ.0 ) GO TO 700
      ROT     = AA(ITYP)/TOC + BB(ITYP) + CC(ITYP)*TOC + DD(ITYP)*TOC**2
      IF( ITYP.EQ.8 ) ROT = 0.88216 * TOC**0.006
      OTE     = XOTE(ITYP)*TOC + YOTE(ITYP) *CLD
C
C
      IF( NSURV.GT.0 ) GO TO 600
      CO3(1) = 0.0
      DO 520 I = 1, 11
      IF( I.GT.1 ) CO3(I) = CO3(I-1) + 0.1
      CO3(11+I) = 0.9
  520 CONTINUE
C
  600 DH      = 0.0
      FM      = 0.2
      H       = 0.0
      CL      = 0.0
      JPASS   = 0
      ITL     = 0
      ITRIM   = 0
C
      CALL OVERLAY(4HOVLY,3,3)
      CALL OVERLAY(4HOVLY,3,4)
C
  700 CONTINUE
C
 1000 FORMAT(1X, *MACH NO. =*F6.3, 5X,*ALTITUDE =*F7.0,* FT.*, 5X,
     1       *SWEEP ANGLE =*F6.2,* DEG.*, 5X,*TAIL DEFL. (DH) =*F6.2,
     2       * DEG.* //T10,*CL*, T22,*TOTAL CD*, T38,*CM*, T53,*ALPHA*,
     3          T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH =0*,
     4        T113,*CD AT DH=0*, / )
 1001 FORMAT(5X,F8.3,2F15.5, F15.2, 4F15.5)
 1002 FORMAT(/10H      CLA =,F8.5,1X,*PER DEG.*,10H      ALO =,F8.5,1X,
     1   *DEG.*,   29X,10H     CLDB =,F8.5 /
     1        10H        K =,F8.5, 9X,        10H   DELCL =,F8.5 ,
     2             34X,10H   CLMAX =,F8.5 /
     2        10H      CMO =,F8.5, 9X,        10H   CM/CL =,F8.4 //
```

```
   3 * DRAG BREAKDOWN ----- *, 8X,*FUSLAGE*, 9X,*BODIES*, 7X,*NACELLES
   4*,11X,*WING*, 5X,*HORIZ TAIL*, 6X,*VERT TAIL*, 7X,*SURFACES* )
 1003 FORMAT(5X,10HFRICTION =,F7.5,7F15.5 /
   1        5X,10HFORM     =,F7.5,7F15.5 /
   2        5X,10HINTERF   =,F7.5,7F15.5 /
   3        5X,10HWAVE     =,F7.5,7F15.5     )
 1004 FORMAT(5X,10HBASE     =,F7.5, 2F15.5 /
   1        5X,10HCAMBER   =,F7.5  /
   2        5X,10HDRAG RISE=,F7.5  /
   3        5X,10HMISC     =,F7.5 )
 1005 FORMAT(1H1, /1X, 6A10 )
 1006 FORMAT(1X, *MACH NO. =*F6.3, 5X,*RN/FT =*1PE15.4,5X,
   1        *L.E. SWEEP ANGLE =* 0PF6.2,* DEG.*,5X,*TAIL DEFL. (DH) =*
   2        F6.2,* DEG.*
   2               //T10,*CL*, T22,*TOTAL CD*, T38,*CM*, T53,*ALPHA*,
   3             T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH=0*,
   4        T113,*CD AT DH=0*, / )
 1007 FORMAT(/5X, 10H   CDMIN =,F7.5 //)
 1008 FORMAT(1X, *MACH NO. =*F6.3, 5X,*ALTITUDE =*F7.0,* FT.*, 5X,
   1        *L.E. SWEEP ANGLE =*F6.2,* DEG.*,5X,*TRIMMED CONDITION*
   2        //T10,*CL*, T22,*TOTAL CD*, T38,*DH*, T53,*ALPHA*,
   3           T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH =0*,
   4         T113,*CD AT DH=0*, / )
 1009 FORMAT(1X, *MACH NO. =*F6.3, 5X,*RN/FT =*1PE15.4,5X,
   1        *L.E. SWEEP ANGLE =* 0PF6.2,* DEG.*,5X,*TRIMMED CONDITION*
   2        //T10,*CL*, T22,*TOTAL CD*, T38,*DH*, T53,*ALPHA*,
   3             T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH=0*,
   4        T113,*CD AT DH=0*, / )
C
      END
```

CC = 00198

```
      SUBROUTINE ADJUST(ID, ID2, XVAR, YVAR)
C
C     ADJUST Y-VARIABLE USING CORRELATION FACTORS
C
      COMMON /BLKADJ/ IVAL(20), X(15), YM(15,9), YA(15,9), XCL(15),
     1                NXVAR, NXCL
C
      COMMON /BLKPRT/ KPRINT(50)
C
      YVAR1  = YVAR
      IF( ID.EQ.0 ) GO TO 300
   50 NV     = IVAL(ID)
C
      IF( NV.EQ.0 ) RETURN
  100 CALL LNTP(XVAR, VM, X,YM(1,NV), NXVAR, 2)
  200 CALL LNTP(XVAR, VA, X,YA(1,NV), NXVAR, 2)
      GO TO 500
C
  300 NV     = IVAL(ID2)
      IF( NV.EQ.0 ) RETURN
  400 CALL LNTP(XVAR, VM, XCL,YM(1,NV), NXCL, 2)
      CALL LNTP(XVAR, VA, XCL,YA(1,NV), NXCL, 2)
C
  500 YVAR   = YVAR1 * VM + VA
C
      IF( KPRINT(24).EQ.0 ) RETURN
      WRITE(6,1000) ID, YVAR1, XVAR, VM, VA, YVAR
 1000 FORMAT(5X,*ADJUST DUMP*, 5X, *ID =*,I5/1X,5F15.5 )
      RETURN
      END
```

CC = 00030

```
      BLOCK DATA
C
      COMMON /BLKBD1/ XSWPL(11), YRMIN(11)
C
      COMMON /BLKBD2/ AAA(22), BBB(22), CCC(22), DDD(22), XTT(22)
C
      COMMON /BLKBD3/ BOT(12),ATSW(6),TPRT(6),FXAC1(216),FXAC2(216)
C
      COMMON /BLKBD4/ XBD4(5), YBD4(5), ZBD4(3), FBD4(5,5,3)
C
      COMMON /BLKBD5/ XDYO(6), YXMN(6), Z1BD5(6,6)
C
      COMMON /BLKBD6/ XBD6(4), YBD6(4), ZBD6(6), FBD61(4,4,6),
     1                FBD62(4,4,6), FBD63(4,4,6)
C
      COMMON /BLKMAP/ MAP, TRANS, DY, AMAP(22), BMAP(11)
      COMMON /BLKT22/ XIN13(10), YIN13(7), ZOUT13(10,7), ZOUT14(10,7)
      COMMON /BLKA15/ NXIN15, XIN15(6),  NYIN15, YIN15(5),
     1                FOUT15(6,5), FOUT16(6,5)
      COMMON /BLKCLB/ X1(6), Y1(6), X9(8), Y9(8), XAR(5), YDCL(5)
      COMMON /BLKA16/ XSWP(7), YTR(4), FEP35(7,4), FEP7(7,4),
     1                XCLDB(7), YAKB(7)
C
C
      COMMON /BLKMAX/ XTR(6), YC1(6), YC2(6), XSWP1(4), YA(4),
     1                YB(4), XDY(8), XM(4), CTAB(8,4), DTAB(8,4),
     2                XXCLM(13), YYDY(6), FCLMX(13,6), XXC2(9),
     3                YYMACH(5), FDCLMX(9,5),XDY1(9),  YXMT(4),
     4                 ZC1MAX(9,4), XDY2(8), YFOC(6), ZDC1M(8,6),
     5                ZZDC1M(8,6)
      COMMON /[illegible]2/ XSP(8), YDYA(6), FDA(8,6), XAB(6), YCO(8),
     1                FKVOFM(6,8), XANG(10), YRTOC(7), FRA(10,7)
      COMMON /[illegible]MX3/ SPAN(6), YTPR(4), FKB(6,4), FKS(6), FKD(6,4)
      COMMON /[illegible]/ XC2(6),YAST(9),FDAM(6,9),XCT(6),YM(5),FDAM2(6,5)
      COMMON /[illegible]X5/ XXXCLM(9), FCLMXX(9,6)
      COMMON /BLKMX6/ XX(12), XY(7), XF(12,7)
      COMMON/BLKMX7/ SPAN2(11), YTPR2(5), FKSW(11,5)
      COMMON/BLKMX8/ SPAN3(6), YTPR3(4), FKM(6,4)
      COMMON /BLKMX9/ SPAN4(5), YBFI1(5), FKA(5,5),
     1                SPAN5(5), YBFI2(3), FKF(5,3)
C
C
C
      DATA XSWPL / 0.0,.17453,.34906,.5236,.65813,.87266,1.04719,
     1             1.13446,1.22173,1.34390,1.57079 /,
     2     YRMIN / 0.53, 0.53, 0.52, 0.515, 0.505, 0.49, 0.45,
     3             0.39, 0.30, 0.12, 0.10 /
C
      DATA BOT  / 0.0, .4, .8, 1.2, 1.6, 1.8, 2., 2.4, 2.8, 3.2,3.6,4./,
     1    ATSW  / 1.,  2., 3., 4., 5., 6. /,
     2    TPRT  / 0., 0.2, 0.25, 0.33, 0.5, 1.0 /
C
C     FIG. 4.1.4.2-22   DATCOM          ****
      DATA FXAC1/.25,.24,.23,.22,.20,.18,.17,.25,.32,.37,.40,.42,
     1          7*0.335, .39, .44, .48, .495, .5,
     2          .42, .44, .45, .46, .465,.475,.50, .56, .56, .57,2*.58,
```

```
     3    .50, .53, .55, .575,.60, .62, 6*0.67,                          R1T1353
     4    .58, .62, .66, .695,.73, .755,.83, .78, .78, .765,.75, .75,    R1T1354
     5    .68, .72, .76, .79, .83, .765,.98, .90, .90, .87, .85, .84,    R1T1355
     6    .29, .27, .25, .25, .23, .215,.20, .28, .33, .39, .44, .46,    R1T1356
     7    .40, .41, .41, .405,.40, .40, .40, .48, .52, .54, .56, .57,    R1T1357
     8    .51, .53, .54, .555,.57, .58, .60, .65, .665,.68, .69, .69,    R1T1358
     9    .64, .66, .68, .70, .73, .76, .795,.80, .81, .815,.805,.80,    R1T1359
     A    .75, .77, .80, .82, .87, .91, .97, .955,.95, .96, .93, .92,    R1T1360
     B    .87, .895,.915,.945,1.0, 1.05,1.15,1.12,1.1, 1.1, 1.05,1.03,   R1T1361
     C    .30, .29, .275,.26, .245,.24, .23, .30, .35, .40, .45, .475,   R1T1362
     D    .41, .42, .425,.425,.42, .415,.41, .50, .54, .57, .595,.60,    R1T1363
     E    .54, .555,.57, .58, .60, .61, .63, .68, .69, .71, .72, .72,    R1T1364
     F    .67, .69, .71, .73, .76, .79, .82, .83, .845,.865,.85, .845,   R1T1365
     G    .80, .82, .84, .86, .90, .95, 1.02,1.01,1.0, 1.01,.98, .97,    R1T1366
     H    .93, .95, .97, .99, 1.05,1.1, 1.25,1.2, 1.16,1.17,1.11,1.1/    R1T1367
      DATA FXAC2 /                                                       R1T1368
     I    .32, .31, .30, .28, .25, .24, .23, .31, .37, .42, .475,.50,    R1T1369
     J    .45, .46, .46, .455,.45, .445,.44, .54, .57, .60, .63, .64,    R1T1370
     K    .595,.60, .61, .62, .64, .65, .665,.72, .74, .76, .78, .775,   R1T1371
     L    .73, .745,.765,.78, .80, .83, .88, .89, .90, .93, .92, .91,    R1T1372
     M    .89, .90, .91, .93, .96, 1.0, 1.07,1.08,1.08,1.1, 1.06,1.05,   R1T1373
     N    1.04,1.05,1.06,1.09,1.12,1.17,1.27,1.26,1.25,1.26,1.21,1.2,    R1T1374
     O    .36, .34, .325,.31, .30, .275,.25, .34, .41, .48, .52, .55,    R1T1375
     P    .52, .52, .52, .52, .51, .505,.50, .60, .63, .67, .71, .72,    R1T1376
     Q    .70, .70, .705,.71, .72, .735,.75, .80, .82, .87, .89, .89,    R1T1377
     R    .88, .885,.89, .90, .915,.94, .98, 1.0, 1.02,1.07,1.05,1.05,   R1T1378
     S    1.04,1.05,1.05,1.06,1.09,1.12,1.19,1.2, 1.22,1.26,1.22,1.21,   R1T1379
     T    1.2, 1.21,1.22,1.24,1.27,1.3, 1.36,1.4, 1.42,1.45,1.4, 1.38,   R1T1380
     U    .50, .48, .46, .44, .40, .38, .34, .42, .50, .60, .70, .76,    R1T1381
     V    .75, .75, .745,.74, .72, .70, .68, .73, .80, .88, .97, 1.0,    R1T1382
     W    5*1.0, .98,.96,1.01,1.08,1.19,1.22,1.23,                       R1T1383
     X    5*1.24,1.23,1.2,1.29,1.38,1.48,1.48,1.49,                      R1T1384
     Y    5*1.49,1.47,1.44,1.55,1.66,1.77,1.73,1.75,                     R1T1385
     Z    3*1.74,1.73,1.71,1.70,1.68,1.81,1.96,2.06,1.98,2.0 /           R1T1386
C                                                                        R1T1387
C  ** FIGURE  4.1.4.1-2 , DATCOM   ******                                R1T1388
      DATA  XBD4 / 0.0, .34907, .69813, .87267, 1.0472 /,                R1T1389
     1      YBD4 / 0.0, 1.5, 3.5, 6.0, 10.0 /,                           R1T1390
     2      ZBD4 / 0.0, 0.5, 1.0 /,                                      R1T1391
     3      FBD4 / 5*0.0, 0.0, 0.0, 0.0, -.0002, -.0005,                 R1T1392
     4             0.0, -.0007, -.002, -.0027, -.0037,                   R1T1393
     5             0.0, -.0019, -.0044, -.006, -.008,                    R1T1394
     6             0.0, -.008, -.016, -.021, -.0265,                     R1T1395
     7             5*0.0, 0.0, 0.0, -.0005, -.0008, -.0011,              R1T1396
     8             0.0, -.001, -.0029, -.0045, -.0068,                   R1T1397
     9             0.0, -.003, -.007, -.0093, -.013,                     R1T1398
     1             0.0, -.008, -.016, -.021, -.0265,                     R1T1399
     2             5*0.0, 0.0, -.0002, -.0006, -.001, -.0012,            R1T1400
     3             0.0, -.0013, -.003, -.004, -.0052,                    R1T1401
     4             0.0, -.0031, -.0066, -.0088, -.012,                   R1T1402
     5             0.0, -.008, -.016, -.0207, -.026   /                  R1T1403
C                                                                        R1T1404
      DATA  XDYO / 0.8, 1.1, 1.4, 1.6, 2.0, 2.4 /,                       R1T1405
     1      YXMN / 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 /,                       R1T1406
     2      Z1BD5/ 3.784,4.395,5.432,6.348,8.545,11.231, 4.029,4.517,    R1T1407
     3             5.249,5.982,7.874,10.254, 4.395,4.639,5.127,5.616,    R1T1408
```

```
     4               7.081,8.973, 4.7,4.834,5.127,5.432,6.287,7.691,      RIT1409
     5               5.371,5.213,5.188,5.31,5.799,6.836,                  RIT1410
     6               6.226,5.738,5.432,5.335,5.616,6.409 /                RIT1411
C                                                                         RIT1412
C                                                                         RIT1413
C     FIGURE 4.1.4.2-25    DATCOM     *****                               RIT1414
      DATA  XBD6 / 0., 1., 2., 3. /,                                      RIT1415
     1      YBD6 / -2., -1., 0., 1. /,                                    RIT1416
     2      ZBD6 / 0., 1., 2., 3., 4., 5. /,                              RIT1417
     3     FBD61 / 0.,.125,.16,.17,0.,.115,.17,.2,0.,.15,.195,.195,0.,    RIT1418
     4 .165,.2,.2,.18,.26,.28,.3,.18,.27,.305,.32,.18,.285,.32,.32,.18,   RIT1419
     5 .3,2*.32,.34,.39,.4,.412,.34,.405,.435,.45,.34,.425,2*.445,.34,    RIT1420
     6.445,.45,.45,.5,.505,.515,.53,.5,.525,.54,.55,.5,.545,2*.55,.5,3*   RIT1421
     7.55, .665,.62,.623,.626,.665,.653,.66,.665,8*.665,                  RIT1422
     8  .815,.73,.728,.722,.815,.775,.778,.78,.815,.79,.78,.78,.815,      RIT1423
     9   3*.775 /                                                         RIT1424
      DATA FBD62 / 0.,.135,.17,.186,0.,.12,.19,.215,0.,.175,2*.215,0.,.2RIT1425
     1,2*.235,.2,.28,.298,.314,.2,.3,.35,.365,.2,.32,2*.36,.2,.365,2*.38RIT1426
     2, .4,.43,.44,.45,.4,.47,.495,.508,.4,.49,2*.51,.4,3*.52,            RIT1427
     3 .6,.575,.578,.58,.6,.61,.63,.65,.6,.64,2*.65,.6,3*.65,             RIT1428
     4 .8,3*.72,.8,.77,.78,.792,.8,.795,.792,.79,.8,3*.78,                RIT1429
     5 .98,.84,.83,.82,.98,3*.92,.98,.925,2*.915,.98,3*.885 /,            RIT1430
     6 FBD63 / 0.,.15,.18,.2,0.,.135,.23,.28,0.,.2,2*.275,0.,.24,2*.28,   RIT1431
     7 .26,.35,.365,.38,.26,.365,.42,.45,.26,.405,2*.455,.26,3*.475,      RIT1432
     8 .5,.535,.55,.565,.5,.585,.61,.63,.5,.605,2*.63,.5,3*.65,           RIT1433
     9 .75,3*.735,.75,.78,.795,.81,.75,3*.8,.75,3*.82,                    RIT1434
     1 .98,.905,.895,.885,.98,3*.95,.98,.95,2*.945,.98,3*.96,             RIT1435
     2 1.17,1.06,1.03,1.,1.17,1.1,1.08,1.065,1.17,1.09,2*1.065,1.17,      RIT1436
     3 3*1.08 /                                                           RIT1437
C                                                                         RIT1438
      DATA XIN13/ 0.0,1., 2., 3., 4., 5., 6., 7., 8., 20. /,              RIT1439
     1     YIN13/ 0.0, 20.0, 35.0, 50.0, 65.0, 72.5, 90.0 /,              RIT1440
     2    ZOUT13/ .00973,.00973,.0093,.009,.0089,.0087,4*0.0086,          RIT1441
     3            .00973,.00973,.0093,.009,.0089,.0087,4*0.0086,          RIT1442
     4            .0097, .0097,.00927,.00902,.0089,.00872,4*.00862,       RIT1443
     5            .0096,.0096,.0092,.009,.0089,.00872, 4*0.00863,         RIT1444
     6            .00943,.00943,.00897,.00882,.00875,.0086,4*.00854,      RIT1445
     7            .0093,.0093,.0088,.0087,.00862,.0085,4*0.00844,         RIT1446
     8            .0093,.0093,.0088,.0087,.00862,.0085,4*0.00844 /        RIT1447
      DATA ZOUT14/1.28,1.28,1.27,1.252,1.244,1.233,1.227,1.217,2*1.21,    RIT1448
     1            1.28,1.28,1.27,1.252,1.244,1.233,1.227,1.217,2*1.21,    RIT1449
     2            1.324,1.324,1.311,1.295,1.281,1.266,1.258,1.24,2*1.23,RIT1450
     3            1.363,1.363,1.35,1.33,1.313,1.292,1.283,1.261,2*1.25,   RIT1451
     4            1.392,1.392,1.38,1.36,1.339,1.317,1.303,1.279,2*1.27,   RIT1452
     5            1.403,1.403,1.392,1.371,1.349,1.325,1.31,1.285,2*1.28,RIT1453
     6            1.403,1.403,1.392,1.371,1.349,1.325,1.31,1.285,2*1.28/RIT1454
C                                                                         RIT1455
      DATA AMAP / 22.0, 21.7, 19.2, 18.35, 22.0, 21.2, 19.2, 27.0,11.75,RIT1456
     1            24.,24.,22.,20.,20.,19.,17.,15.,29.,27.,25., 2*0. /,    RIT1457
     2     BMAP / 4 * 1.75, 3 * 2.0, 4 * 0.0 /                            RIT1458
C                                                                         RIT1459
      DATA NXIN15/ 6 /,                                                   RIT1460
     1      XIN15/ 0.0, 0.5, 0.6, 0.8, 0.9, 1.0 /,                        RIT1461
     2      NYIN15/ 5 /,                                                  RIT1462
     3      YIN15 / 0.0, 5.0, 15.0, 20.0, 23.0 /,                         RIT1463
     4      FOUT15/ 0.865, 0.756, 0.71, 0.6, 0.51, 0.45,                  RIT1464
```

```
     5            0.865, 0.756, 0.71, 0.6, 0.51, 0.45,                      R1T1465
     6            0.925, 0.965, 0.958,.905,.845, 0.78,                      R1T1466
     7            0.978, 1.02,1.03, 1.02, 0.99, 0.975,                      R1T1467
     8            0.99,  1.04,1.05, 1.05, 1.045,1.04 /,                     R1T1468
     9       FOUT16/ 6 * 1.05, 1.05, 1.04, 1.02,.94,.88,.84,                R1T1469
     1             1.05,  1.05, 1.04, 1.01, 0.995, 0.985,                   R1T1470
     2            6 * 1.05,  6 * 1.05 /                                     R1T1471
      DATA X1    / 0.7243, 0.8727, 1.0472, 1.2217, 1.3963, 1.5708 /,        R1T1472
     1     Y1    / 0.85, 0.69, 0.575, 0.51, 0.483, 0.48 /,                  R1T1473
     2     X9    / 0.3665,0.5236,0.6981,0.8727,1.0472,1.2217,1.3963,1.6/,R1T1474
     3     Y9    / 0.85,.625,.435,.35,0.375,0.447,0.478, 0.48 /,            R1T1475
     4     XAR   / 1.0, 2.3, 3.0, 4.0, 5.0 /,                               R1T1476
     5     YDCL  / -0.16, 0.0, 0.05, 0.085, 0.0855 /                        R1T1477
C                                                                           R1T1478
      DATA  AAA / 8.48243E-4, 8.03417E-4, 3.31322E-6, -1.72323E-3,          R1T1479
     4            2.02978E-3, 1.36146E-4, 3.79206E-5, 15 * 0.0 /,           R1T1480
     5      BBB / -.0296429, -.0349676, -.0142049, .0310185,                R1T1481
     6            -.07305, -4.05767E-3, -1.03545E-3, 15 * 0.0  /,           R1T1482
     7      CCC / 1.23835, 1.15999, 1.03825, 0.585837, 1.57142, .712684,R1T1483
     8            .639394, 0.0, 0.0, 5 * 1.1019, 3 * .275475,               R1T1484
     9            3 * 3.3057, 0.0, 0.0 /,                                   R1T1485
     1      DDD / -2.47002, -1.71448, -1.9292, -.559968, -2.99739,          R1T1486
     2            .0188291, .0607728, 15 * 0.0 /,                           R1T1487
     3      XTT / .35, .375, .41, .45, .37, .39, .42, .371, .5, .2,         R1T1488
     4            .3, .4, .5, .6, .3, .4, .5, .3, .4, .5, 0.0, 0.0 /        R1T1489
      DATA  XSWP  / 0.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0 /,              R1T1490
     1      YTR   / 0.0, 0.25, 0.5, 1.0 /,                                  R1T1491
     2      FFP35 / 0.94,0.951,0.96,0.968,0.973,0.974,0.971,                R1T1492
     3              0.992,0.995,0.997,0.999,0.998,0.992,0.984,              R1T1493
     4              1.0,1.0,0.999,0.997,0.988,0.971,0.944,                  R1T1494
     5              0.996,0.992,0.986,0.975,0.959,0.93,0.863 /,             R1T1495
     6      FEP7  / 0.89,0.907,0.923,0.936,0.944,0.948,0.942,               R1T1496
     7              0.99,0.994,0.996,0.998,0.991,0.978,0.956,               R1T1497
     8              0.996,0.992,0.987,0.98,0.961,0.924,0.86,                R1T1498
     9              0.986,0.968,0.942,0.91,0.874,0.833,0.79 /               R1T1499
      DATA  XCLDB / 0.0, 0.3, 0.4, 0.5, 0.6, 0.7, 2.0 /,                    R1T1500
     1      YAKB  / 0.33,0.352,0.44,0.625,1.0,1.47, 2.0 /                   R1T1501
C                                                                           R1T1502
C                                                                           R1T1503
      DATA XTR     / 0.0, 0.1, 0.2, 0.3, 0.5, 1.0 /,                        R1T1504
     1     YC1     / 0.0,0.225,0.47,0.5,0.32, 0.0 /,                        R1T1505
     2     YC2     / 0.0,0.22,0.5, 0.91, 1.05, 0.85 /,                      R1T1506
     3      XSWP1   / 0.0, 30.0, 50.0, 60.0             /,                  R1T1507
     4     YA      / 0.9, 1.04, 1.2, 1.3             /,                     R1T1508
     5     YB      / 0.0, 0.24, 0.5, 0.71            /,                     R1T1509
     6     XDY     / 0.0, 2.0, 2.25, 2.5, 3.0, 4.0, 4.5, 10.0 /,            R1T1510
     7     XM      / 0.2, 0.4, 0.6, 1.0              /                      R1T1511
      DATA CTAB    / 8 * 0.0, -.02,-.02,-.26,-.32,-.355,-.41,-.445,-.45, R1T1512
     9               .12,.12, -.36, -.43, -.5, -.64, -.72, -.72,            R1T1513
     1               .58,.58, -.15, -.34, -.57, -.76, -.92, -.92 /,         R1T1514
     2     DTAB    / 8 * 0.0, -.085,-.085, .02, .055, .045, .02,.02,.02, R1T1515
     3               -.2, -.2, -.01, -.02, -.03, -.07, -.09, -.09,8*0.0/,R1T1516
     4     XXCLM   / 0.0, .2, .4, .6, .8, 1., 1.2, 1.6, 2.0, 2.4, 2.8,     R1T1517
     5               3.2, 4.4 /,                                            R1T1518
     6     YYDY    / 0.0, .25, .5, .75, 1.0, 10.0 /                         R1T1519
      DATA FCLMX   / 0.9, 1.375, 1.57, 1.645, 1.645, 1.575, 1.4, 1.14,     R1T1520
```

```
     8              1.02, .96, .92, .91, .9,   .75, 1.28, 1.48, 1.55,    R1T1521
     9              1.56, 1.5, 1.33, 1.11, 1.0, .94, .9, .88, .875,      R1T1522
     1              .68, 1.19, 1.4, 1.47, 1.475, 1.4, 1.27, 1.08,        R1T1523
     2              .975, .92, .88, .85, .84,    .6, 1.1, 1.3, 1.375,    R1T1524
     3              1.385, 1.33, 1.23, 1.06, .96, .9, .865, .84, .83,    R1T1525
     4              .4,1.02, 1.2, 1.29, 1.31, 1.27, 1.19, 1.04, .94,     R1T1526
     5              .88, .84, .83, .82,    .4, 1.02, 1.2, 1.29, 1.31,    R1T1527
     6              1.27, 1.18, 1.02, .91, .85, .82, .8, .8    /,        R1T1528
     7      XXC2   / 0.0, 2.0, 4., 6., 8., 10., 12., 14., 20./,         R1T1529
     8      YYMACH / 0.0, 0.2, 0.4, 0.6, 1.0   /,                        R1T1530
     9      FDCLMX / -.11, -.08, 0.0, .115, .23, .335, .36, .22, 0.0,    R1T1531
     1              -.11, -.08, 0.0, .115, .23, .335, .36, .22, 0.0,     R1T1532
     2              -.11, -.08, 0.0, .1, .2, .3, .32, .18, 0.0,          R1T1533
     3              -.11, -.08, 0.0, .08, .15, .21, .205, .08, 10*0.0/   R1T1534
C                                                                        R1T1535
C  ** FIGURE  4.1.1.4-5,   DATCOM    *****                               R1T1536
C                                                                        R1T1537
      DATA  XDY1  / 0.0, 1.1, 2.25, 2.5, 3.0, 3.5, 4., 4.5, 6. /,       R1T1538
     1       YXMT  / 0.3, 0.35, 0.4, 0.45 /,                             R1T1539
     2      ZC1MAX / 0.8, 0.8, 1.315,1.43,1.58,1.58,1.54,1.47,1.41,      R1T1540
     3              0.8, 0.8, 1.315,1.43,1.51,1.52,1.49,1.41,1.36,       R1T1541
     4              0.8, 0.8, 1.315,1.39,1.45,1.45,1.43,1.35,1.34,       R1T1542
     5              0.8, 0.8, 1.3,  1.33,1.35,1.36,1.33,1.32,1.32/,      R1T1543
     6       XDY2  / 0.0,1.0,2.0,2.2,2.4,3.0,5.2,10.0 /,                 R1T1544
     7       YFDC  / 0.0, 2.0, 4.0, 6.0, 8.0, 100.0 /,                   R1T1545
     8      ZDC1M  / 10*0.0, 0.1, 0.13, 0.15, 0.145,0.05, 0.0,           R1T1546
     9              0.0, 0.11,0.275,0.3,0.3,0.22,0.10,0.0,               R1T1547
     1              0.1,0.26,0.44,0.465,0.45,0.30,0.17,0.0,              R1T1548
     2              0.2,0.40,0.60,0.620,0.60,0.39,0.25,0.0,              R1T1549
     3              0.2,0.40,0.60,0.620,0.60,0.39,0.25,0.0 /             R1T1550
      DATA Z2DC1M/ 9*0.,.15,.28, .28, .27, .18, .07, .07,                R1T1551
     1             .1, .28, .49, .47, .41, .26, .12, .12,                R1T1552
     2             .2, .41, .61, .60, .55, .35, .17, .17,                R1T1553
     3             .33,.52, .68, .65, .60, .39, .20, .20,                R1T1554
     4             .33,.52, .68, .65, .60, .39, .20, .20 /               R1T1555
C                                                                        R1T1556
      DATA XSP   / 0.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 90.0/,       R1T1557
     1      YDYA / 0.0, 1.2, 2.0, 3.0, 4.0, 50.0 /,                      R1T1558
     2      FDA  / 1.8, 2.2, 3.4, 5.0, 7.4, 10.2, 13.4, 13.4,            R1T1559
     3             1.8, 2.2, 3.4, 5.0, 7.4, 10.2, 13.4, 13.4,            R1T1560
     4             0.1, 1.1, 2.3, 3.9, 5.8, 6.75, 9.9 , 9.9 ,            R1T1561
     5             1.3, 1.7, 2.4, 3.2, 4.2, 5.3 , 6.7 , 6.7 ,            R1T1562
     6             2.2, 2.0, 2.1, 2.3, 2.5, 2.85, 3.25, 3.25,            R1T1563
     7             2.2, 2.0, 2.1, 2.3, 2.5, 2.85, 3.25, 3.25 /           R1T1564
C                                                                        R1T1565
      DATA  XAB   / 0.0, 1.0, 2.0, 3.0, 4.0, 6.0 /,                      R1T1566
     1       YCO   / -1., -.75, -.5, -.25, 0.0, .25, .5, .75 /,          R1T1567
     2     FKVOFM  / 1.57, 1.74, 2.02, 2.30, 2.58, 2.75,                 R1T1568
     3              1.78, 1.95, 2.18, 2.52, 2.85, 2.90,                  R1T1569
     4              2.03, 2.21, 2.42, 2.71, 3.00, 3.15,                  R1T1570
     5              2.48, 2.60, 2.73, 2.93, 3.19, 3.33,                  R1T1571
     6              3.13, 3.15, 3.20, 3.28, 3.45, 3.60,                  R1T1572
     7              3.79, 3.80, 3.81, 3.82, 3.88, 3.98,                  R1T1573
     8              6 * 4.45, 6 * 5.0 /                                  R1T1574
      DATA  XANG  / 0.0,.0698,.1396,.2094,.2792,.3490,.4188,.4886,       R1T1575
     1              0.5584, 0.6980 /,                                    R1T1576
```

```
     2      YRTOC  / 0.0, 0.03, 0.05, 0.08, 0.12, 0.16, 0.5 /,             R1T1577
     3      FRA / 0.45,0.245,0.075, 7*0.0,                                 R1T1578
     4            1.0,.62,.40,.255,.19,.148,.11,.079,.05, 0.0,             R1T1579
     5            1.0,.84,.72,.48,.34,.252,.195,.142,.1,  0.0,             R1T1580
     6            1.0,.90,.858,.758,.618,.46,.345,.25, .17, 0.0,           R1T1581
     7            1.0,.917,.885,.814,.72,.62,.51,.38, .23, 0.0,            R1T1582
     8            1.0,.925,.895,.835,.76,.665,.565, .46, .35, 0.0,         R1T1583
     9            1.0,.925,.895,.835,.76,.665,.565, .46, .35, 0.0 /        R1T1584
C                                                                          R1T1585
      DATA  SPAN / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,                         R1T1586
     1      YTPR / 0.0, 0.25, 0.5, 1.0 /,                                  R1T1587
     2      FKB  / 0.0, 0.3, 0.555, 0.77, 0.925, 1.0,                      R1T1588
     3             0.0, 0.28,0.53, 0.75, 0.915, 1.0,                       R1T1589
     4             0.0, 0.265, 0.51, 0.73, 0.9, 1.0,                       R1T1590
     5             0.0, 0.25, 0.48, 0.70, 0.885, 1.0 /,                    R1T1591
     6      FKS  / 0.0, 0.31, 0.63, 0.85, 0.945, 1.0 /,                    R1T1592
     7      FKD  / 0.0, 0.38, 0.64, 0.84, 0.965, 1.0,                      R1T1593
     8             0.0,0.295, 0.54, 0.74, 0.9,   1.0,                      R1T1594
     9             0.0, 0.24, 0.475, 0.68, 0.855, 1.0,                     R1T1595
     1             0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /                          R1T1596
C                                                                          R1T1597
      DATA  XC2  / 0.0, 1.0, 2.0, 3.0, 4.0, 5.0 /,                         R1T1598
     1      YAST / 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 9.0, 30.0 /,         R1T1599
     2      FDAM / 10.,7.,4.,1.5,0.,.5, 8.6,5.5,2.5,.5,-.3,.5,             R1T1600
     3             7.5,4.,1.,-.6,-1.,5, 5.5,1.5,-1.2,-2.5,-1.,.5,          R1T1601
     4             3.,-1.5,-4.1,-3.,-1.,.5, .2,-4.5,-5.,-3.,-1.,.5,        R1T1602
     5             -2.3,-6.5,-5.,-3.,-1.,.5, -4.5,-7.,-5.,-3.,-1.,.5,      R1T1603
     6             -8.,-7.,-5.,-3.,-1.,0.5  /,                             R1T1604
     7      XCT  / 4.0, 6.0, 8.0, 10.0, 12.0, 14.0 /,                      R1T1605
     8      YM   / 0.2, 0.4, 0.6, 0.8, 1.0 /,                              R1T1606
     9      FDAM2 / 0.,1.5,4.,7.2,11.,15., 0.,.8,3.,6.,9.5,13.5,           R1T1607
     1             0.,.3,1.5,3.2,5.5,8.6, 0.,0.,.5,2.,4.,7.,               R1T1608
     2             0., .8, 3., 6., 9.5, 13.5 /                             R1T1609
C                                                                          R1T1610
      DATA XXXCLM / 0.0, .2, .4, .6, .8, 1., 1.2, 1.6, 2.0 /,              R1T1611
     1      FCLMXX / 0.7,1.2,1.37,1.43,1.43,1.365,1.25,1.08,.99,           R1T1612
     1             .65,1.12,1.29,1.355,1.365,1.305,1.21,1.07,.98,          R1T1613
     2             .6,1.09,1.23,1.29,1.295,1.24,1.17,1.05,.97,             R1T1614
     3             .55,1.0,1.15,1.23,1.24,1.20,1.15,1.04,.96,              R1T1615
     4             .5,.91,1.09,1.17,1.20,1.175,1.13,1.02,.935,             R1T1616
     5             .5,.91,1.09,1.17,1.20,1.165,1.11,1.0,.91 /              R1T1617
C                                                                          R1T1618
      DATA  XX   / 0.0,.4,.8,1.0,1.2,1.4,1.6,2.0,2.4,2.8,3.2,3.37 /,       R1T1619
     1      XY   / 0.,.1309,.1745,.2618,.3490,.4363,1.0 /,                 R1T1620
     2      XF   / 8*1.0,.915,.63,.29,0.0,                                 R1T1621
     3             8*1.0,.915,.63,.29,0.0,                                 R1T1622
     4             7*1.0,.915,.755,.52,.245,0.0,                           R1T1623
     5             1.0,.97,.95,.97,1.0,.95,.87,.69,.552,.38,.185,0.0,      R1T1624
     6             .92,.90,.90,.95,1.0,.95,.845,.63,.48,.31,.150,0.0,      R1T1625
     7             .79,.80,.84,.91,1.0,.945,.83,.585,.425,.27,.128,0.,     R1T1626
     8             .79,.80,.84,.91,1.0,.945,.83,.585,.425,.27,.128,0. /    R1T1627
C                                                                          R1T1628
      DATA SPAN2 /0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0 /,           R1T1629
     1      YTPR2 / 0.0, 0.2, 0.333, 0.5, 1.0 /,                           R1T1630
     2      FKSW  / 0.0, 0.0285, 0.0435, 0.0490, 0.0480, 0.0417,           R1T1631
     3         0.0320, 0.0212, 0.0112, 0.0035, 0.0,                        R1T1632
```

```
4            0.0, 0.0310, 0.0460, 0.0545, 0.0565, 0.0530,          RIT1633
5         0.0460, 0.0350, 0.0240, 0.0120, 0.0,                     RIT1634
6            0.0, 0.0290, 0.0460, 0.0560, 0.0590, 0.0570,          RIT1635
7         0.0510, 0.0410, 0.0290, 0.0150, 0.0,                     RIT1636
8            0.0, 0.0270, 0.0450, 0.0560, 0.0610, 0.0600,          RIT1637
9         0.0545, 0.0450, 0.0330, 0.0180, 0.0,                     RIT1638
1            0.0, 0.0225, 0.0400, 0.0525, 0.0605, 0.0620,          RIT1639
2         0.0600, 0.0520, 0.0400, 0.0230, 0.0 /                    RIT1640
C                                                                  RIT1641
 DATA SPAN3 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,                      RIT1642
1     YTPR3 / 0.0, 0.25, 0.5, 1.0 /,                               RIT1643
2     FKM   / 0.0, 0.645, 1.045, 1.240, 1.320, 1.330,              RIT1644
3             0.0, 0.440, 0.760, 0.955, 1.070, 1.120,              RIT1645
4             0.0, 0.310, 0.575, 0.775, 0.930, 1.030,              RIT1646
5             0.0, 0.200, 0.400, 0.600, 0.800, 1.000  /           RIT1647
 DATA SPAN4 /  0.0, 0.4, 0.8, 1.6, 2.4 /,                          RIT1648
1     YBFI1 /  0.0, 0.05, 0.1, 0.15, 0.2 /,                        RIT1649
2     FKA   /  0.0, 0.38, 0.68, 1.00, 1.13,                        RIT1650
3              0.0, 0.40, 0.74, 1.12, 1.32,                        RIT1651
4              0.0, 0.49, 0.83, 1.38, 1.75,                        RIT1652
5              0.0, 0.57, 1.05, 1.81, 2.43,                        RIT1653
6              0.0, 0.70, 1.27, 2.34, 3.37 /                       RIT1654
 DATA SPAN5 /  0.2, 0.4, 0.6, 0.8, 1.0 / ,                         RIT1655
1     YBFI2 /  0.0, 0.1, 0.2 /,                                    RIT1656
2     FKF   /  4.00, 1.60, 0.60, 0.17, 0.0,                        RIT1657
3              3.40, 1.55, 0.65, 0.26, 0.12,                       RIT1658
4              2.96, 1.46, 0.72, 0.40, 0.25 /                      RIT1659
 END                                                               RIT1660

                                                    CC = 00364
```

```
      OVERLAY(3,1)
      PROGRAM VGEOM
C
C     COMPUTES GEOMETRY FOR VARIABLE-SWEEP CONFIGURATIONS
C
      COMMON /BLKVGM/ KPASS, SWPLE, DOBR, CLDR, ARWR, SWLER, SWQCR,
     1                SWMCR, SWTER, TOCR, TANSR, DZ, SXR, SWPMCR,
     2                CBXR, CTXR, CBXPR, SOXPR, ARXRR, AROPR, FOCR,
     2                DA1, DAC1, DSWP1, CPOCR,
     3                DA2, DAC2, DSWP2, DTOC
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKA02/ SREF, AR, TAPR, SWPR,
     1                BLEN(10), BWID(10), BHGT(10), BAWET(10), BQ(10),
     2                BNO(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),
     3                BASE(10),  ELEN(10), EWID(10), EHGT(10),
     4                EAWET(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),
     5                ELBT(10),  EQF(10), ENO(10), CBAR(10), TW,
     6                XLEW, YWW, YBD,CR, BO2, BFUS, FMISC, AB, AFTAW,
     6                CAM(10), TOC(10), AWET(10), SWMT, SPLAN, CONCL,
     7                TWIST, ETWIST, WINC, XLE(11), CRW(11), YW(11),
     8                XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC,
     9                SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),
     1                SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),
     2                SCR(10), HTLE, HTY, HTZ, HTINC
C
      COMMON /BLKG01/ FRBOD(10),ARS(10),SSEX(10),FRNAC(10),SWP,SWPQC,
     1                SWPMC, SWPTE, DOB, TOCW, CLD, SEXW,  ESWMC,
     2                SXX(10), XLESWX(10), XMCSWX(10), ARW, TR, WPLAN,
     3                CB, YB, XB, CRX, CBX, CTX, YIX, YOX, SIX, SOX,
     4                ARI, ARXR, CBXP, AROP, SOXP, SWPLEI, SWPLEC,
     5              SMPMCI, SWPMCO, ESWQC, ESWLE, ESWTE, CROB, DXOB,
     6              XHT, XCRTE, XH, OMEGA, FOC, TWIST, DINC, SWPMT,
     7                CBAR2, CLDS, TOCS, SWET, DXQC, G1(81)
C
      COMMON /BLKPRT/ KPRINT(50)
      DIMENSION X(6), Y(6), XT(6), YT(6), D(6), P(6)
C
C
C
      IF( KPASS.EQ.1 ) GO TO 5
      IF(SWPLE.EQ.SWPR) GO TO 10
      IF( KPASS.EQ.2 ) GO TO 20
      GO TO 25
    5 DOBR   = DOB
      CLDR   = CLD
      SWLER  = ESWLE
      SWQCR  = ESWQC
      SWMCR  = ESWMC
      SWTER  = ESWTE
      ARWR   = ARW
      TOCR   = TOCW
      CBXR   = CBX
      CTXR   = CTX
      CBXPR  = CBXP
```

```
      SOXPR   = SOXP
      ARXRR   = ARXR
      AROPR   = AROP
      FOCR    = FOC
      SXR     = SXX(NPNLS)
      SWPMCR  = XMCSWX(NPNLS)
      GO TO 15
C
C
   10 DOB     = DOBR
      TR      = TAPR
      CLD     = CLDR
      CBAR2   = CBAR(NPNLS)
      TWIST   = TWIST
      DINC    = WINC
      SWPMT   = SWMT
      WPLAN   = SPLAN
      ARW     = ARWR
      CLDS    = CAM(NPNLS)
      TOCS    = TOC(NPNLS)
      SWET    = AWET(NPNLS)
      DXQC    = XLE(1+NPNLS)+CRW(1+NPNLS)*0.25   -XLE(1) -CRW(1)*0.25
      CLD     = CLDR
      DOB     = DOBR
      TOCW    = TOCR
      SXX(NPNLS) = SXR
      XLESWX(NPNLS)= SWPR
      XMCSWX(NPNLS)= SWPMCR
      CBX     = CBXR
      CTX     = CTXR
      SOX     = SXR
      YOX     = YW(1+NPNLS) - YW(NPNLS)
      CBXP    = CBXPR
      SOXP    = SOXPR
      ARXR    = ARXRR
      AROP    = AROPR
      SWPLEO  = SWPR
      SWPMCO  = SWPMCR
      SEXW    = SXX(1) + SXX(2)
C
      ESWLE   = SWLER
      ESWQC   = SWQCR
      ESWMC   = SWMCR
      ESWTE   = SWTER
      GO TO 50
C
   15 BO2     = YW(1+NPNLS)
      CR      = SPLAN/(1.+TAPR) /BO2
      CRX2    = CR * (1.- YW(NPNLS)/BO2 *(1.-TAPR) )
      DZ      = CR * TAPR * TAN(TWIST/57.296)
      TANSR   = TAN(SWPR) * (1.- (1.-TAPR)*CR/BO2/TAN(SWPR) )
      SX2     = (CRX2 + CR*TAPR) * (YW(1+NPNLS) - YW(NPNLS) )
      SWET    = SX2 * (2.+.1843*TOC(NPNLS) +1.5268*TOC(NPNLS)**2
     1                -.8395*TOC(NPNLS)**3)
      DA1     = AWET(NPNLS) - SWET
      DAC1    = CBAR(NPNLS) -.66667 *(CRX2 + (CR*TAPR)**2/
```

```
     1                          (CRX2 + CR*TAPR) )
      SWPMC  = ATAN(0.5 *(TANSR + TAN(SWPR)) )
      DSWP1  = SWMT - ATAN(0.5 *(TAN(SWPR) + TAN(SWPMC)) )
      SWPTE  = ATAN(TANSR)
      CPOCR  = 0.5 * (COS(SWPR)/COS(SWPR-SWPMC) + COS(SWPTE)/
     1                                   COS(SWPMC-SWPTE) )
C
      DA2    = 0.0
      DAC2   = 0.0
      DSWP2  = 0.0
      DTOC   = 0.0
C
      GO TO 50
C
   20 CONTINUE
C
C
C
   25 Y(2)   = YW(1+NPNLS)
      CR     =  SPLAN/(Y(2)*  (1.+TAPR))
      X(4)   = CR + XAPEX
      X(1)   = XAPEX
      Y(1)   = 0.0
      Y(4)   = 0.0
      X(2)   = X(1) + Y(2) * TAN(SWPR)
      X(3)   = X(2) + TAPR * CR
      Y(3)   = Y(2)
      DO 30 I =1,4
      D(I)    = SQRT((X(I) - XPIVOT)**2 + (Y(I) - YPIVOT)**2)
      P(I)    = ATAN((X(I) - XPIVOT) / ( Y(I) - YPIVOT) )
      IF((Y(I) - YPIVOT).LE.0.0) P(I) = P(I) + 3.141592
   30 CONTINUE
C
      DSWP    = SWPLE - SWPR
C
      DO 40 I =1,4
      XT(I)   = XPIVOT +D(I) * SIN(DSWP + P(I))
      YT(I)   = YPIVOT +D(I) * COS(DSWP + P(I))
   40 CONTINUE
      DXQC    = 0.75* XT(2) +0.25* XT(3) -XLE(1) - CRW(1)*0.25
      SWPTE   = ATAN((XT(3) - XT(4))/(YT(3) - YT(4)))
      CO      = TAN(SWPTE) / TAN(SWPLE )
      SWPQC   = ATAN((1.0 - (1.0 - CO)/4.0)* TAN(SWPLE) )
      SWPMC   = ATAN((1.0 - (1.0 - CO)/2.0)* TAN(SWPLE) )
      XFWD      = XT(2) - YT(2) * TAN(SWPLE)
      XAFT      = XT(3) - YT(3) * TAN(SWPTE)
      CRCLP   = XAFT - XFWD
      CPOC    = 0.5 * ( COS(SWPLE)/COS(SWPLE - SWPMC) + COS(SWPTE)/
     1                                        COS(SWPMC - SWPTE) )
      CLDS    = CAM(NPNLS) * CPOC/CPOCR
      TOCS    = TOC(NPNLS) * CPOC/CPOCR
      DOB     = DOBR * Y(2)/YT(2)
C
   45 CONTINUE
      TTIP    = 1.0 + YT(2)* TAN(SWPLE) * (CO - 1.0)/CRCLP
      TWIST   = ATAN(DZ/(CRCLP*TTIP)) * 57.296
```

```
      DINC    = WINC * (1.-TAN(DSWP) * TANSR) * COS(DSWP)
C
      C3      = CRCLP * (1.0 -(YT(3)/YT(2))*(1.0 - TTIP) )
      WPLAN   = (CRCLP + C3) *YT(3) + C3 *(YT(2)-YT(3))
      ARW     = 4. * YT(2)**2/WPLAN
      YMID    = YT(3) +(YT(2) - YT(3)) /(1.0/SQRT(C3/(TTIP*CRCLP)) + 1.0)
      TR      = 1. - (YMID/YT(2)) * (1.-TTIP)
      XTE2    = XT(4) + YW(NPNLS)*TAN(SWPTE)
      CRX2    = CRCLP * (1.-YW(NPNLS)/YT(2)  * (1.-TTIP))
      SX2     = WPLAN - (CRCLP + CRX2) * YW(NPNLS)
      STOTAL  = SX2
      TOCW    = TOCS
      CLD     = CLDS
      IF( NPNLS.EQ.1 ) GO TO 46
C
      S1      = (XT(4)-XT(3))/(YT(4)-YT(3))
      S2      = (XLE(1)+CRW(1)-XLE(2)-CRW(2))/(YW(1)-YW(2))
      IF( S1.EQ.S2 ) GO TO 46
      YI      =(XT(4)-XLE(1)-CRW(1) +S1*YT(4) -S2*YW(1))/(S1-S2)
      DA      =(XTE2 - XLE(2) - CRW(2)) * (YW(2) - YI)
      SX2     = SX2 + DA
      STOTAL  = SXX(1) + SX2
      TOCW    = SQRT((TOC(1)**2*SXX(1) + TOCS**2*SX2)/STOTAL)
      CLD     = SQRT((CAM(1)**2*SXX(1) + CLDS**2*SX2)/STOTAL)
C
   46 SWET    = SX2 * (2.+.1843*TOCS +1.5268*TOCS**2 -.8395*TOCS**3)
     1          +DA1 +(DA2-DA1) * (SWPLE-SWPR)/(AFTSW-SWPR)
      TP      = TR * CRCLP/CRX2
      CBAR2   = 0.66667 * CRX2 * (1. + (TP**2/(1.+TP)) )
      CBAR2   = CBAR2 + DAC1 + (DAC2 - DAC1) * DSWP/(AFTSW-SWPR)
C
      SWPMT   = SWPQC + DSWP1 + (DSWP2-DSWP1) * DSWP/(AFTSW-SWPR)
      TOCW    = TOCW + DTOC * DSWP/(AFTSW - SWPR)
C
      COSMC   = COS(XMCSWX(1)) * SXX(1) + COS(SWPMC) * SX2
      COSQC   = COS(ATAN(0.5*(TAN(XLESWX(1))+TAN(XMCSWX(1))))) * SXX(1)
     1                + COS(SWPQC) * SX2
      TANLE   = TAN(XLESWX(1)) * SXX(1) + TAN(SWPLE) * SX2
      TANTE   = (2.*TAN(XMCSWX(1))- TAN(XLESWX(1))) *SXX(1)
     1                + TAN(SWPTE) * SX2
C
      ESWMC   = ACOS(COSMC/STOTAL)
      ESWQC   = ACOS(COSQC/STOTAL)
      ESWLE   = ATAN(TANLE/STOTAL)
      ESWTE   = ATAN(TANTE/STOTAL)
C
      SEXW    = STOTAL
      SXX(NPNLS) = SX2
      XLESWX(NPNLS)= SWPLE
      XMCSWX(NPNLS)= SWPMC
      CBX     = CRX2
      CTX     = CRCLP * TR
      SOX     = SX2
      YOX     = YMID - YW(NPNLS)
      CBXP    = CRCLP * (1.-(YW(1)+YW(2))*.5 *(1.-TR)/YMID )
      SOXP    = (CBXP + CTX) * (YMID -(YW(1)+YW(2))*.5)
```

```
      ARXR   = 4. * (YMID - YW(1))**2 /SEXW
      AROP   = 4. * (YMID - (YW(1)+YW(2)*.5) )**2/SOXP
      SWPLEO = SWPLE
      SWPMCO = SWPMC
      FOC    = CLD * FOCR /CLOR
C
      IF( KPASS.NE.2) GO TO 50
C
      IF( AFTAW.GT.0.0 ) DA2 = AFTAW - SWET
      IF( AFTCB.GT.0.0 ) DAC2 = AFTCB - CBAR2
      IF( AFTOC.GT.0.0 ) DTOC = AFTOC - TOCS
   50 CONTINUE
      SWP    = SWPLE
C
      IF( KPRINT(12).EQ.1 ) WRITE(6,1000) KPASS, SWPLE, DA1, DAC1,
     1        DSWP1, CPOCR, DA2, DAC2, DSWP2, DTOC, DA
     2          ,TWIST,DINC,SWPMT,CBAR2,CLDS,TOCS,SWET,DOB,TOCW,
     3           CLD,SEXW,ESWLE,ESWQC,ESWMC,ESWTE,ARW,TR,WPLAN
 1000 FORMAT(5X, *VGEOM DUMP,   KPASS =*,I3, 10X,*SWPLE =* F12.4 /
     1          (1X, 7F15.5 )
C
      END
```

CC = 00246

```
      OVERLAY(3,2)
      PROGRAM MCRIT
C
C     COMPUTES CRITICAL MACH NUMBER
C
      COMMON /BLKOV3/ ID, XMACH, TOC, CLD, SWEEP, CV3(3), JPASS,
     1                OV3A(3)
      COMMON /BLKA05/ SCW(6), SECTG(5), CIMAX, CLMCR(10), XMCR(10),
     1                XT(33), YT(33), YC(33), NMCR, NXSET
C
      COMMON /BLKC03/ CLTAB(11), TABMCR(11)
C
      IF( ID.LE.23 ) CALL CPZT(ID, 0.6, TOC, CLD, SWEEP)
C
  100 CLTAB(1) = 0.0
      DO 200 I = 1, 11
      IF( I.GT.1 ) CLTAB(I) = CLTAB(I-1) + 0.1
      CALL LNTP(CLTAB(I), TABMCR(I), CLMCR, XMCR, NMCR, 2)
      CALL ADJUST(0,4, CLTAB(I), TABMCR(I))
      IF( TABMCR(I).LT.0.0 ) TABMCR(I) = 0.0
C
  200 CONTINUE
C
  300 WRITE(6,1000) (CLTAB(I), TABMCR(I) , I =1,11)
C
 1000 FORMAT(1H1, 5X, * MACH CRITICAL TABLE * /// 5X *  CL  *, 5X
     1                * MACH CRITICAL * //(5X,2F15.4) )
C
      END
```

CC = 00029

```
      SUBROUTINE CPZT(ID, XMACH, TOC, CLD, SWEEP)
C
C     CALCULATES AIRFOIL PRESSURES - BRITISH METHOD
C     AND COMPUTES MACH CRITICAL USING  CREST CRITERIA
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKA05/ SCW(6), SECTG(5), CIMAX, CLMCR(10), XMCR(10),
     1                ORD(99), NMCR, NXSET
      COMMON /BLKR07/ XU(32), ZT(32), ZS(32), ROC, ZTE
      COMMON /BLKCPI/ SA(32), SB(32), SC(32), SD(32), SE(32)
      COMMON /BLKPRT/ KPRINT(50)
      COMMON /BLKG01/ G1(200)
C
      DIMENSION COSE(32), SINE(32), S1(32,32), S2(32,32), S3(33,32),
     1                   S4(32,32), S5(32,32), YCPI(32), DZDXUT(32)
C
      PI     = 3.1415927
      AR     = G1(80)
      SWP    = SWEEP/57.29578
      COSPHI = ( COS(SWP) )**(AR/(1.4+AR))
      IF( SWEEP.GT.40.0 ) COSPHI = 0.5 * (COSPHI +.76604**(AR/(1.4+AR)))
C     COSPHI IS THE COSINE OF THE EFFECTIVE ISOBAR SWEEP
      SWP    = ACOS(COSPHI)
C
      NCP    = 31
      NMCR   = 0
C
      DO 100 I= 1,NCP
C
      XI     = I
      XN     = NCP + 1
      NX      = NCP + 1
      COSE(I)= COS(XI*PI/XN)
      SINE(I)= SIN(XI*PI/XN)
      XU(I)  = 0.5*(1.0 + COSE(I))
  100 CONTINUE
C
      CALL SECT(ID, TOC, CLD)
C
      DO 300 IV = 1, NCP
      DO 200 IU = 1, NCP
C
      IMV     = IU - IV
      S1(IU,IV) = XN/SINE(IU)
      S2(IU,IV) = COSE(IU)/(SINE(IU)**2)
      S3(IU,IV) = XN/SINE(IU)
      S4(IU,IV) = XN/SINE(IU) - 2.0 * ((-1.0)**IU -1.0) /
     1               ( XN * SINE(IU) * (1.0 - COSE(IU)) )
      S5(IU,IV) = - S2(IU,IV)
      IF( IU.EQ.IV ) GO TO 150
C
      SIGN      = ((-1.0)**IMV - 1.0)/XN
      DEM       = (COSE(IU) - COSE(IV))
      S1(IU,IV) = SIGN * 2.0 * SINE(IU) /(DEM**2)
      S2(IU,IV) = (-2.0*(-1.0)**IMV*SINE(IU))/(SINE(IV)*DEM)
```

```
      S3(IU,IV) = S1(IU,IV) - SIGN * 2.0 * (1.0/(SINE(IU)*DEM))
      S4(IU,IV) = 2.0 * SIGN * (1.0 - COSE(IU)*COSE(IV) ) /
     1            (SINE(IV) * DEM**2 )
     2          - 2.0 * ((-1.0)**IU -1.0) /
     3            (XN * SINE(IV) * (1.0 - COSE(IU)) )
      S5(IU,IV) = -2.0 * (-1.0)**IMV /DEM
C
  150 IF( KPRINT(26).NE.3 ) GO TO 200
      WRITE(6,1000) S1(IU,IV), S2(IU,IV), S3(IU,IV)
     1              ,S4(IU,IV), S5(IU,IV)
  200 CONTINUE
      S3(NX,IV) = ((-1.0)**IV -1.0)/(XN*(1.0 + COSE(IV)))
  300 CONTINUE
C
      IF( KPRINT(26).EQ.3 ) WRITE(6,1000)  (S3(NX,J), J=1,NCP)
C
      ALPHA  = 6.0
  350 A      = ALPHA/57.296
      NMCR   = NMCR + 1
      XM     = XMACH
      IF( XM.GT.0.9 ) XM = 0.9
      IF(XM.LE.0.0 ) XM = 0.01
      XM2     = XM**2
      IF( KPRINT(1).EQ.1 ) WRITE(6,1001) XM, ALPHA, SWEEP
      CLI    = 0.0
      CL     = 0.0
      DCPIM1 = 0.0
      DCPM1  = 0.0
      CMI    = 0.0
      CM     = 0.0
      DO 500 IV = 1, NCP
      SA(IV)    = 0.0
      IF(ID.LT.10.OR.ID.GT.20) SB(IV) = 0.0
      SC(IV)    = S3(NX,IV) * SQRT(ROC/2.0)
      SD(IV)    = 0.0
      SE(IV)    = 0.0
      DO 400 IU = 1, NCP
C
      SA(IV)    = SA(IV) + S1(IU,IV)*ZT(IU)
      IF(ID.LT.10.OR.ID.GT.20) SB(IV) = SB(IV) +S2(IU,IV)*ZT(IU)
      SC(IV)    = SC(IV) + S3(IU,IV)*ZT(IU)
      SD(IV)    = SD(IV) + S4(IU,IV)*ZS(IU)
      SE(IV)    = SE(IV) + S5(IU,IV)*ZS(IU)
  400 CONTINUE
C
      CALL CPUOV(1.0, A, SWP, IV, CPIU, CPU, XM)
C
      CALL CPUOV(-1., A, SWP, IV, CPIL, CPL, XM)
C
      DCPI   = CPIL - CPIU
      DCP    = CPL  - CPU
C
      DZDXU  = SB(IV) + SE(IV)
      DZDXL  =-SB(IV) + SE(IV)
C
      YCPI(IV) = CPIU
```

```
      DZDXUT(IV) = DZDXU
C
      IF( KPRINT(1).EQ.1 ) WRITE(6,1000) XU(IV),CPIU,CPIL,DCPI,DZDXU,
     1                         DZDXL, CPU, CPL, DCP, XU(IV)
C
      DX      = XU(NCP)
      XB      = 0.5 * (1. + XU(1))
      IF( IV.NE.1 )  DX = XU(IV-1) - XU(IV)
      IF( IV.NE.1 ) XB = 0.5 * (XU(IV-1) + XU(IV))
      CLI     = CLI + DX * (DCPI + DCPIM1) * 0.5
      CL      = CL  + DX * (DCP  + DCPM1 ) * 0.5
      CMI     = CMI + DX * (DCPI + DCPIM1)*0.5 * (XB - 0.25)
      CM      = CM  + DX * (DCP + DCPM1) *0.5 * (XB - 0.25)
C
      DCPIM1 = DCPI
      DCPM1  = DCP
C
  500 CONTINUE
C
      CLI     = CLI + XU(NCP) * DCPI* 0.5
      CL      = CL  + XU(NCP) * DCP * 0.5
      CMI     = CMI + XU(NCP) * DCPI * 0.5 * (0.5*XU(NCP) -0.25)
      CM      = CM  + XU(NCP) * DCP * 0.5 * (0.5*XU(NCP) -0.25)
      CPCRIT = (1.42857/XM2) * (0.52828 *(1. + 0.2 *XM2)**3.5 -1.)
C
      IF( KPRINT(1).GT.0 ) WRITE(6,1002) CLI,CMI,CPCRIT,ZTE,CL,CM
  600 CONTINUE
C
      IF( A.LT.DZDXUT(NCP) ) GO TO 610
      NMCR    = NMCR - 1
      ALPHA   = ALPHA -1.
      GO TO 350
C
  610 CALL LNTP(A, XCREST, DZDXUT, XU, NCP, 4)
C
  620 CALL LNTP(XCREST, CPCR, XU, YCPI, NCP, 4)
C
      CPCRST = CPCR/COSPHI**2
  630 CPI2   = CPCRST * CPCRST
      XMDDN  = 1./(1.0323 -.8365*CPCRST -.361*CPI2 -.1336*CPCRST**3
     1                 -.0173*CPCRST**4)
      IF( ID.EQ.8 )  XMDDN = 1./(1.012 -.8551*CPCRST
     1            -.4493*CPI2 -.2219*CPCRST**3 -.0447*CPCRST**4)
C
      XMDD   = XMDDN/COSPHI
      IF( XMDD.LE.0.82.OR.ID.EQ.8 ) GO TO 640
      IF( XMDD.LE.1.08 ) XMDD = .2686 -.5729*XMDD +2.7123*XMDD**2
     1                                 -1.4582*XMDD**3
      IF( XMDD.GT.1.08 ) XMDD = 0.8545 +.1132*XMDD
C
  640 FMCRB  = 1.0
      IF( NBODYS.GE.1 ) FMCRB = 1.0439 - 0.5828/G1(1)
      IF( G1(1).GT.10.0 ) FMCRB = 0.98
      IF( XMDD.GT.FMCRB ) XMDD = FMCRB
C
      BMDD2  = 1. - (XMDD * COS(SWP))**2
```

```
      IF( BMDD2.GT.0.0 ) CLMDD = CLI/SQRT(BMDD2)                        R1T2107
C                                                                       R1T2108
      IF( KPRINT(1).GT.0 ) WRITE(6,1003) XMDD,CLMDD,XCREST,CPCR,COSPHI  R1T2109
      CLMCR(NMCR) = CLMDD                                               R1T2110
      XMCR(NMCR)  = XMDD                                                R1T2111
C                                                                       R1T2112
  700 IF( CLI.LE.0.0 ) RETURN                                           R1T2113
      IF( NMCR.GE.10 ) RETURN                                           R1T2114
      ALPHA  = ALPHA - 1.                                               R1T2115
      GO TO 350                                                         R1T2116
C                                                                       R1T2117
 1000 FORMAT(5X,10F12.6)                                                R1T2118
C                                                                       R1T2119
 1001 FORMAT(1H1, ////10X, 57HPRESSURE DISTRIBUTION AND SURFACE SLOPES FR1T2120
     1OR 2-D AIRFOIL   , 6X,*MACH =*F6.4, 5X,*ALPHA =*F6.3 ///          R1T2121
     1            10X,*SWEEP ANGLE =*, F7.3 //                          R1T2122
     2      12X,*X/C*,7X,*CPI(UPPER)*,2X,*CPI(LOWER)*,4X,*DCPI*,        R1T2123
     3      5X,*DZ/DX(UPPER)*,2X,*DZ/DX(LOWER)*,1X,*CP(UPPER)*,         R1T2124
     4      2X,*CP(LOWER)*,4X,*DCP*, 9X,*X/C* // )                      R1T2125
C                                                                       R1T2126
 1002 FORMAT(//10X,*CLI =*F7.4, 10X,*CMI =*F7.4, 20X,*CP(M=1) = *,      R1T2127
     1      F8.4, 10X, *ZTE =*F7.4 /11X,*CL =*F7.4, 6X,*CM(C/4) =*      R1T2128
     2      F7.4 / )                                                    R1T2129
C                                                                       R1T2130
 1003 FORMAT( /10X,*MACH CRIT =*F7.4, 10X, *CL AT MDD =*F7.4, 10X,      R1T2131
     1          *X-CREST =*F7.4, 10X,*CP-CREST =*F8.4 /                 R1T2132
     2          50X, *COSINE OF EFFECTIVE ISOBAR =*F7.4 //)             R1T2133
C                                                                       R1T2134
      END                                                               R1T2135

                                                     CC = 00197
```

```
      SUBROUTINE SECT(ID, TOC, CLD)
C
C     CALCULATES THICKNESS AND CAMBER
C
      COMMON /BLKCPI/ SA(32), SB(32), SC(32), SD(32), SE(32)
      COMMON /BLKR07/ XU(32), ZT(32), ZS(32), ROC, ZTE
      COMMON /BLKA05/ XMU, ZMU, XML, ZML, ZPTE, ZTHIK, RLE, A4(25),
     1                XT(33), YT(33), YC(33), NMCR, NXSET
      COMMON /BLKBD2/ AA(22), BB(22), CC(22), DD(22), XTT(22)
      COMMON /BLKPRT/ KPRINT(50)
C
      DIMENSION XSECT(26),Y164A(26),CAMB(26)
      DIMENSION Y163A(26),Y165A(26),Y163(26),Y164(26),Y165(26),
     1          Y166(26),CAMB1(26)
      DIMENSION A0(11), A1(11), A2(11), A3(11), D1(11), D2(11), D3(11)
C
      DATA XSECT/ 0.0,0.005,0.0075,0.0125,0.025,0.05,0.075,0.1,
     1            0.15,0.2,0.25,0.3,0.35,0.4,0.45,0.5,0.55,0.6,
     2            0.65,0.7,0.75,0.8,0.85,0.9,0.95,1.0 /
      DATA Y163 / 0.0,.00829,.01004,.01275,.01756,.0244,.0295,.03362,
     1            .03994,.04445,.04753,.04938,.05,.04938,.04766,.04496,
     2            .0414,.03715,.03234,.02712,.02166,.01618,.01088,
     3            .00604,.00214,0.0 /
      DATA Y164 / 0.0,.0082,.00989,.0125,.01701,.02343,.02826,.03221,
     1            .03842,.04302,.04639,.04864,.0498,.04988,.04843,
     2            .04586,.04238,.0382,.03345,.02827,.02281,.01722,
     3            .01176,.00671,.00248,0.0 /
      DATA Y165 / 0.0,.00772,.00932,.01169,.01574,.02177,.02647,.0304,
     1            .03666,.04143,.04503,.0476,.04924,.04996,.04963,
     2            .04812,.0453,.04146,.03682,.03156,.02584,.01987,
     3            .01385,.0081,.00306,0.0 /
      DATA Y166 / 0.0,.00759,.00913,.01141,.01516,.02087,.02536,.02917,
     1            .0353,.04001,.04363,.04636,.04832,.04953,.05,.04971,
     2            .04865,.04665,.04302,.03787,.03176,.02494,.01773,
     3            .01054,.00408,0.0 /
      DATA Y163A/ 0.0,.00816,.00983,.0125,.01737,.02412,.02917,.03324,
     1            .0395,.044,.04714,.04913,.04995,.04968,.04837,.04613,
     2            .04311,.03943,.03517,.03044,.02545,.0204,.01535,.0103,
     3            .00525,.00021 /
      DATA Y164A/ 0.0,0.00804,0.00969,0.01225,0.01688,0.02327,
     1            0.02815,0.03199,0.03813,0.04272,0.04606,0.04837,
     2            0.04968,0.04995,0.04894,0.04684,0.04388,0.04021,
     3            0.03597,0.03127,0.02623,0.02103,0.01582,0.01062,
     4            0.00541,0.00021 /
      DATA Y165A/ 0.0,.00765,.00928,.01183,.01623,.02182,.0265,.0304,
     1            .03658,.04127,.04483,.04742,.04912,.04995,.04983,
     2            .04863,.04632,.04304,.03899,.03432,.02912,.02352,
     3            .01771,.01188,.00604,.00021 /
      DATA CAMB / 0.0,0.00281,0.00396,0.00603,0.01055,0.01803,
     1            0.02432,0.02981,0.03903,0.04651,0.05257,0.05742,
     2            0.06120,0.06394,0.06571,0.06651,0.06631,0.06508,
     3            0.06274,0.05913,0.05401,0.04673,0.03607,0.02452,
     4            0.01226,0.0  /
      DATA CAMB1/ 0.0,.0025,.0035,.00535,.0093,.0158,.0212,.02585,
     1            .03365,.0398,.04475,.0486,.0515,.05355,.05475,.05515,
     2            .05475,.05355,.0515,.0486,.04475,.0398,.03365,.02585,
```

```
     3              .0158,0.0 /
      DATA A0   / 5 * 0.2969, 3 * 0.14845, 3 *0.514246 /,
     1      A1  / 0.213337, -.096082, -.246867, -.310275, -.27118,
     2            0.412103, 0.193233, 0.083362, -.840115, 2*0.0    /,
     3      A2  / -2.931954, -.54331, 0.175334, 0.3417, 0.1402,
     4            -1.67261, -.558166, -.18315,  1.1101,  2*0.0    /,
     5      A3  / 5.22917, 0.559395, -.266917, -.32182, -.082137,
     6                 1.68869, .283208, -.00691, -1.09401, 2*0.0  /,
     7      D1  / 0.2,  0.234, 0.315, 0.465, 0.7, 0.234, 0.315,
     8            0.465, 0.234, 0.315, 0.465                      /,
     9      D2  / -.040625, -.068571, -.233333, -.684, -1.6625,
     1            -.068571, -.233333, -.684, -.068571, -.23333, -.684/,
     2      D3  / -.070312, -.093878, -.032407, 0.292, 1.3125,
     3            -.093878, -.032407, 0.292, -.093878, -.032407, 0.292/
C
      NCP     = 31
      ROC     = AA(ID) + BB(ID)*TOC +CC(ID)*TOC*TOC + DD(ID)*TOC**3
C
C     ID  = 1 TO 4, 6  SERIES AIRFOIL           IJ = 1
C     ID  = 5 TO 7, 6A SERIES AIRFOIL           IJ = 1
C     ID  = 8, SUPERCRITICAL  AIRFOIL           IJ = 2
C     ID  = 9, BICONVEX                         IJ = 3
C     ID  = 10 TO 20, 4 DIGIT AIRFOIN           IJ = 4
C     ID  = 21,22, INPUT AIRFOIL                IJ = 5
C
      IF( ID.LE.7 ) IJ = 1
      IF( ID.EQ.8 ) IJ = 2
      IF( ID.EQ.9 ) IJ = 3
      IF( ID.GE.10.AND.ID.LE.20 ) IJ = 4
      IF( ID.GE.21 ) IJ = 5
C
      GO TO (200,300,100,900,700) , IJ
C
  100 WRITE(6,2000)
      DO 110 I = 1,NCP
      ZT(I)     = 2,0*TOC*XU(I)*(1.0 - XU(I))
      SB(I)   = 2.*TOC * (1.-XU(I))
      ZS(I)     = 0.0
  110 CONTINUE
      ROC       = 0.0
      GO TO 960
C
  200 IF( ID.LE.4 ) WRITE(6,2003)
      IF( ID.GT.4 ) WRITE(6,2001)
      DO 220 I = 1,NCP
      X         = XU(I)
      IF( ID.LE.4 ) CALL LNTP(X, ZS1, XSECT, CAMB1, 26, 4)
      IF( ID.GT.4 ) CALL LNTP(X, ZS1, XSECT, CAMB,  26, 4)
      ZS(I)     = ZS1 * CLD
  210 IF( ID.EQ.1 ) CALL LNTP(X, ZT(I), XSECT, Y163,  26, 4)
      IF( ID.EQ.2 ) CALL LNTP(X, ZT(I), XSECT, Y164,  26, 4)
      IF( ID.EQ.3 ) CALL LNTP(X, ZT(I), XSECT, Y165,  26, 4)
      IF( ID.EQ.4 ) CALL LNTP(X, ZT(I), XSECT, Y166,  26, 4)
      IF( ID.EQ.5 ) CALL LNTP(X, ZT(I), XSECT, Y163A, 26, 4)
      IF( ID.EQ.6 ) CALL LNTP(X, ZT(I), XSECT, Y164A, 26, 4)
      IF( ID.EQ.7 ) CALL LNTP(X, ZT(I), XSECT, Y165A, 26, 4)
```

```
C                                                                       R1T2249
  220 ZT(I)     = ZT(I) * TOC/0.1                                       R1T2250
      GO TO 960                                                         R1T2251
C                                                                       R1T2252
  300 WRITE(6,2002)                                                     R1T2253
      WRITE(6,1000) XMU,ZMU, XML,ZML, ZPTE, ZTHIK                       R1T2254
  400 CONTINUE                                                          R1T2255
C                                                                       R1T2256
C     * * * * * * * * * * * * * * * * * * * * * * * * * * *             R1T2257
C                                                                       R1T2258
C     CARDS 2257 THRU 2335 ARE NOT SHOWN IN THIS LISTING                R1T2259
C                                                                       R1T2334
C     * * * * * * * * * * * * * * * * * * * * * * * * * * *             R1T2335
C                                                                       R1T2336
      GO TO 960                                                         R1T2337
C                                                                       R1T2338
C                                                                       R1T2339
  700 ROC       = RLE                                                   R1T2340
C                                                                       R1T2341
      DO 800 I = 1, NCP                                                 R1T2342
      X         = XU(I)                                                 R1T2343
      CALL LNTP(X, ZT(I), XT, YT, NXSET, 4)                             R1T2344
      CALL LNTP(X, ZS1,   XT, YC, NXSET, 4)                             R1T2345
      ZS(I)     = ZS1 * CLD                                             R1T2346
  800 CONTINUE                                                          R1T2347
      GO TO 960                                                         R1T2348
C                                                                       R1T2349
  900 WRITE(6,2004)                                                     R1T2350
      I4D       = ID - 9                                                R1T2351
      ROC       = 0.5 * (AO(I4D)*TOC/.2)**2                             R1T2352
C                                                                       R1T2353
      DO 950 I =1, NCP                                                  R1T2354
      X         = XU(I)                                                 R1T2355
      IF( X.GT.XTT(ID) ) GO TO 910                                      R1T2356
C                                                                       R1T2357
      ZT(I)     = AO(I4D) * SQRT(X) + A1(I4D) * X  + A2(I4D) * X*X       R1T2358
     1                    + A3(I4D) * X**3                              R1T2359
      SB(I)     = 0.5 * AO(I4D)/SQRT(X) +A1(I4D) +2.*A2(I4D)*X          R1T2360
     1                 + 3.* A3(I4D) * X**2                             R1T2361
      GO TO 940                                                         R1T2362
C                                                                       R1T2363
  910 X         = (1. - X)                                              R1T2364
      ZT(I)     = 0.002    + D1(I4D) * X + D2(I4D) * X*X  + D3(I4D) * X**3  R1T2365
      SB(I)     = -D1(I4D) -2.* D2(I4D)*X  -3.* D3(I4D)*X*X             R1T2366
C                                                                       R1T2367
  940 ZT(I)     = TOC/.2 * ZT(I)                                        R1T2368
      SB(I)     = TOC/.2 * SB(I)                                        R1T2369
      CALL LNTP(X, ZS1, XSECT, CAMB1, 26, 4)                            R1T2370
      ZS(I)     = ZS1 * CLD                                             R1T2371
C                                                                       R1T2372
  950 CONTINUE                                                          R1T2373
C                                                                       R1T2374
  960 IF( KPRINT(1).EQ.0 ) RETURN                                       R1T2375
      WRITE(6,1004) TOC, CLD, ROC                                       R1T2376
      DO 970 I = 1, NCP                                                 R1T2377
      INV       = NCP + 1 - I                                           R1T2378
```

```
      ZUP     = ZT(INV) + ZS(INV)
      ZLO     =-ZT(INV) + ZS(INV)
      WRITE(6,1005) XU(INV), ZT(INV), ZS(INV), ZUP, ZLO
  970 CONTINUE
C
 1000 FORMAT(/5X,*XMU =*,F10.6,3X,*ZMU =*,F10.6,3X,*XML =*,F10.6,
     1             3X,*ZML =*F10.6, /
     1            3X,*ZPTE =*F10.6,3X,*ZTHIK =*,F10.6 // )
 1001 F3RMAT(5X, 7F10.6 / )
 1002 FORMAT( I5,5X,5F10.0 )
 1003 FORMAT( 10F6.0 )
 1004 FORMAT(10X, *T/C =*F7.4, 10X,*CAMBER =*F7.4, 10X,*L.E.RADIUS =*,
     1        F8.5 //T17,*X/C*,T26,*THICKNESS*,T44,*CAMBER*,T60,
     2        *UPPER*,T75,*LOWER*   )
 1005 FORMAT(5X, 5F15.5 )
 2000 FORMAT(/ 10X, *BICONVEX AIRFOIL SECTION*/ )
 2001 FORMAT(/10X, *6A SERIES AIRFOIL SECTION * /)
 2002 FORMAT(/ 10X, *SUPERCRITICAL AIRFOIL SECTION*  /)
 2003 FORMAT( /10X, *6 SERIES AIRFOIL SECTION* /)
 2004 FORMAT( /10X, *4-DIGIT AIRFOIL SECTION* /)
      RETURN
      END
```

CC = 00190

```
      SUBROUTINE CPUOV(S, A, SWP, IV, CPI, CP, XM)
C
C     COMPUTES CP FOR AN INFINITELY SHEARED WING
C
      COMMON /BLKCPI/ SA(32), SB(32), SC(32), SD(32), SE(32)
C
      COMMON /BLKR07/ XU(32), ZT(32), ZS(32), RCC, ZTE
C
   10 COSA    = COS(A)
      SINA    = SIN(A)
      COSL    = COS(SWP)
      SINL    = SIN(SWP)
C
   20 F1      = SA(IV)
      F2      = SB(IV)
      F3      = SC(IV)
      F4      = SD(IV)
      F5      = SE(IV)
      FX      = SQRT((1. - XU(IV))/XU(IV))
   30 DUL     = 1.0/(1. +((F2 +S* F5)/COSL)**2 )
C
   40 UIOV2   = DUL * ( COSA * (1. + F1 * COSL +S* F4 * COSL)
     1            +S* SINA * COSL * (1. + F3/COSL) * FX )**2
     2          + DUL * ( COSA * SINL * (F1 +S* F4)
     3            +S* SINA * SINL * (1. + F3/COSL) * FX )**2
     4          + (SINL * COSA)**2 * (1. - DUL)
C
      CPI     = 1.0 - UIOV2
C
      CP      = CPI
C
   50 CPIO    = (1. - SINL * SINL + F1 * F1 + 2.0 * F1 * COSL)
     1            /(1. + (F2/COSL)**2 )    - SINL * SINL
      CPIO    = 1.0  - CPIO
      IF( CPIO.GT.0.0 ) CPIO = 0.0
C
      XMN     = XM * COSL
      IF( XM.LE.0.01.OR.XMN.GE.1.0 ) RETURN
C
      XM2     = XM * XM
      BETA    = SQRT(1. - XMN * XMN)
C
      SINA    = SINA/BETA
   60 TEST    = (COSL * COSL - CPIO * XMN) * XM2
      IF( TEST.GE.1.0 ) WRITE(6,1000)
      IF( TEST.GE.1.0 ) GO TO 200
C
   70 B       = SQRT(1.0 - TEST)
C
      F1      = F1/B
      F4      = F4/BETA
      F3      = F3/B
   80 DUL     = 1.0/(1. + ((F2 +S* F5)/(B * COSL))**2 )
C
   90 UOV2    = DUL * ( COSA * (1. + F1 * COSL +S* F4 * COSL)
     1            +S* SINA * COSL * (1. + F3/COSL) * FX )**2
```

```
     2        + DUL * ( COSA * SINL * (F1 +S* F4)                              R1T2458
     3           +S* SINA * SINL * (1. + F3/COSL) * FX )**2                     R1T2459
     4        + (SINL * COSA)**2 * (1. - DUL)                                  R1T2460
C                                                                              R1T2461
  100 CP      = -1.42857/XM2                                                   R1T2462
C                                                                              R1T2463
      TEST    = 1.0 + 0.2 * XM2 * (1. - UOV2)                                  R1T2464
      IF( TEST.LE.0.0 ) GO TO 200                                              R1T2465
      CP      = -CP * ( TEST**3.5 - 1.0 )                                      R1T2466
C                                                                              R1T2467
  200 CONTINUE                                                                 R1T2468
      RETURN                                                                   R1T2469
 1000 FORMAT(10X,*KUCHEMAN-WEBER CORRECTION FACTOR REACHES LIMIT* )            R1T2470
      END                                                                      R1T2471

                                                          CC = 00070
```

```
      OVERLAY(3,3)
      PROGRAM AERO
C
C     AERODYNAMICS
C
      COMMON /BLKOV3/ ITYP, OV3A(4), ALT, SPEED, SWEEP, JPASS, OV3B(2),
     1                ITRIM
C
      COMMON /BLKG01/ G1(200)
C
      COMMON /BLKA02/ A1(433)
C
      COMMON /BLKC01/ CL, CD, CM, ALPHA, CDM, CDL, CDR, CDRO, CLT, CDT,
     1                DH, FK, DELCL, CMO, DCMCL, XACWB, CLA, ALO,
     2                C1(45), CLDB, CLMAX, C2(7), FML1, FML2, CDC,
     3                C3(12), XACS, C4(6), CDAFT, C5(5)
C
C
      CB      = G1(83)
C
      RNOFT = ABS(ALT) * 10.0**6
      IF( ALT.LT.0.0) GO TO 80
      CALL ATMOS(ALT,T,SIGMA,RHO,THETA,DELTA,CA,AMU,1)
      RNOFT  = SPEED*CA*RHO/AMU
   80 CONTINUE
C
   70 FMACH=SPEED
C
C
      IF( JPASS.EQ.0 ) CALL CDDR(0.0, FMACH, RNOFT, CDRO)
      IF( JPASS.EQ.1 ) CALL CDDR1(0.0, FMACH, RNOFT, CDRO)
      CALL CDDR1(CL, FMACH, RNOFT, CDRCL)
      CDR     = CDRCL - CDRO
      IF( SPEED.GT.FML1 ) CDR = 0.0
C
      IF( JPASS.EQ.2 ) GO TO 30
C
      CALL DMIN(FMACH, RNOFT, CDMIN)
      CALL CLWBT(FMACH)
      RE      = RNOFT * CB
      IF( RE.GT.4.0E6 ) RE = 4.0E6
      RE      = RE/1.0E6
C
      CALL CLBRK(FMACH, RE, RNOFT)
      CALL CDL1(FMACH, RNOFT, FK, DELCL, PRIMEK, AKD, AKB)
      CALL CDRG(FMACH, FK, DELCL, CDC)
      CALL CMOW(FMACH, CMO)
      CALL ADJUST(1,0, FMACH, CMO)
C
      CALL WBAC(FMACH, XACWB)
C
   30 CONTINUE
      CALL CDL2(FMACH, CL, FK, DELCL, PRIMEK, AKD, AKB, CDL)
      IF( JPASS.LE.1 ) CALL AERA(FMACH, CL, ALPHA)
      CALL AERA1(FMACH, CL, ALPHA)
      IF( JPASS.GT.0 ) GO TO 50
```

```
      IF( A1(241).EQ.0.0 ) GO TO 50
C
      DO 40 I = 1, 21
      ANG     = I - 1
      CALL AFTCD(ANG, DCD)
      IF ( I.EQ.1 ) WRITE(6,1000)
   40 WRITE(6,1001)  ANG, DCD
   50 CALL AFTCD(ALPHA, CDAFT)
      CALL TDRG(ITRIM, FMACH, DCLT, DCDT)
C
      CDM     = CDMIN + CDC + CDRO
      CD      = CDM + CDL + CDR + CDAFT
C
      CDT = CD + DCDT
      CLT = CL + DCLT
C
C
      JPASS  = 2
C
 1000 FORMAT( /// 10X, *FUSLAGE AFT-END UPSWEEP DRAG * //
     1              10X, *WING ANGLE*, 10X, *DELCD*  / )
 1001 FORMAT( 1X, 2F15.5 )
      END
```

CC = 00079

```
      SUBROUTINE CDDR(CL, XMACH, RNOFT, CDR)
C
C     COMPUTES DRAG RISE
C
      COMMON /BLKC03/ CLTAB(11), TABMCR(11)
C
      COMMON /BLKC01/ C1(18), TCD(5), C2(48), FMCRO, FML1, FML2, CDC,
     1                CLAMCR, CLAML2, C3(12), PO, A3, PL, C4(8)
C
      COMMON /BLKG01/ G1(200)
C
      COMMON /BLKPRT/ KPRINT(50)
C
      ESWMC  = G1(49)
      TOC    = G1(46)
      FOC    = G1(111)
C
      RNOFT1 = RNOFT / XMACH
      CALL LNTP(0.0, FMCRO, CLTAB, TABMCR, 11, 2)
C
      CALL FORG(1.0, RNOFT1)
      CDFF   = TCD(2) + TCD(3)
C
      CALL WORG(1.00001)
      CDW1   = TCD(4)
C
      CALL WORG(1.01)
      CDW2   = TCD(4)
C
      CDWP   = (CDW2 - CDW1) * 100.
      CDW1   = CDW1 - CDFF
C
      XM     = 1.-FMCRO
C
      A3     = (XM * CDWP -2.* CDW1)/XM**3
      A2     = (3.* CDW1 - XM * CDWP)/XM**2
      PL     = 25. * (TOC + 2.* FOC) * COS(ESWMC)**3
      PO     = A2 - PL
C
C
      FML1   = 0.95
      IF( FMCRO.GT.0.90 ) FML1 = FMCRO   .05
      IF( FML1.GT.1.0 ) FML1 = 1.0
      FML2   = FML1 + 0.15
C
      RNOFT2 = RNOFT*FMCRO/XMACH
      CALL FORG(FMCRO,RNOFT2)
      CALL WORG(FMCRO)
      CALL CLWBT(FMCRO)
      CLAMCR = C1(17)
C
      RNOFT2 = RNOFT*FML2/XMACH
      CALL FORG(FML2,RNOFT2)
      CALL WORG(FML2)
      CALL CLWBT(FML2)
      CLAML2 = C1(17)
```

```
C
      IF( KPRINT(23).EQ.1 ) WRITE(6,1000) FMCRO, A2, A3, FML1, FML2,
     1                 CLAMCR, CLAML2, CDW1, CDFF, CDWP
C
C
      ENTRY CDDR1
      CDR     = 0.0
      CDRO    = 0.0
      CALL LNTP(CL, XMCR, CLTAB, TABMCR, 11, 2)
      IF( XMACH.GT.1.0  ) RETURN
      XM      = XMACH - XMCR
      XMO     = XMACH - FMCRO
      IF( XMO.GT.0.0 ) CDRO = PO * XMO**2 + A3 * XMO**3
      IF( XM.GT.0.0 )  CDR  = PL * XM**2
      IF( XM.GT.0.12 ) CDR = 0.0144*PL + 0.24*PL *(XM -.12)
      CDR     = CDR + CDRO
C
      RETURN
 1000 FORMAT(5X, *CDDR DUMP* /(1X, 7F15.5))
      END
```

CC × u '076

```
      SUBROUTINE CLWBT(SPEED)
C
C     COMPUTES WING-BODY-TAIL LIFT CONTRIBUTION
C
      COMMON /BLKG01/ G1(44), DOB, TOC, CLD, SEXW, ESWMC, G2(30),
     1                AR, TR, SPLAN, G3(118)
C
      COMMON /BLKA01 / NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKA02 / SREF, A1(432)
C
      COMMON /BLKC01/ C1(16), CLA, ALO, C2(35), CLAW, CLAB, CLAT, A, B,
     1                AOH, C, ABREAK, CLDH, CLPB, CLDB, CLMAX, ABRK,
     2                AMAX, DAMAX, DEL, CLS, ARLO, C3(29)
C
      COMMON /BLKCLA/ CLAI(11)
C
C
      CLAW   = 0.0
C
      CLAI(1) = SPLAN
      CLAI(2) = TOC
      CLAI(3) = TR
      CLAI(4) = AR
      CLAI(5) = 0.0
      IF( A1(234).EQ.8. ) CLAI(5) = 0.0334
      CLAI(6) = CLD
      CLAI(7) = 0.0
      IF( A1(234).EQ.8. ) CLAI(7) = 0.09
      CLAI(8) = DOB
      CLAI(9) = 0.0
      IF( A1(234).EQ.8. ) CLAI(9) = 1.173763 * TOC
      CLAI(10)=ESWMC
      CLAI(11)= SREF
C
      CALL AER2(SPEED, CLA)
      CLAW   = CLA
C
      CLAT   = 0.0
      A      = 0.0
      B      = 0.0
      C      = 0.0
      ABREAK = 90.0
C
      IF( NHT.EQ.0 ) GO TO 30
C
   20 CALL TAIL(SPEED)
   30 CONTINUE
C
      CLAB= 0.0
      IF (NBODYS.EQ.0) GO TO 50
C
      WB     = A1(14)
      HB     = A1(24)
      AB     = A1(64)
      BL     = A1(4)
```

```
      BLN    = A1(84)
      PERIM = 3.14159*SQRT((WB**2 + HB**2)/2.)
      BHPF = WB/HB*PERIM/SQRT(AB)
      AK1    = 0.009942*BHPF -.000379*BHPF**2 +.00001*BHPF**3
      CLAB   = AK1 * (BL/BLN)**0.3333 * AB/SRE
C
   50 CLA = CLAW + CLAT + CLAB
C
      CALL AALO(SPEED)
      CALL ADJUST(3,0, SPEED, ALO)
C
      RETURN
      END
```

CC = 00069

```
      SUBROUTINE AER2(SPEED, CLA)
C
C     LIFT CURVE SLOPE
C
      COMMON /BLKPRT/ KPRINT(50)
      COMMON /BLKCLA/ SPLAN, TOC, TAPER, ARWS, DMSTR, CLD, DM1, DOB,
     1                EPSL, SWPMC, SREF
C
C
C
      COSZ  = COS(SWPMC)
      PI    = 3.14159
      TWOD   = 1.0
      IF( SPEED.LT.1.0 ) TWOD = 1. + EPSL/SQRT(1.-SPEED**2)
C
C
      CSUBO  = (10.0 + 0.91 * ARWS**3)/(10.0 + ARWS**3)
      ZMSTR  = CSUBO + (1.0 -CSUBO) * (1.0 - COSZ)**2
      ZMSTRO  = ZMSTR
      ZMSTR = ZMSTR + DMSTR
      ZM1    = 1.0 -2.0 * TOC * (ARWS**3/(4.0 +ARWS**3))
     1          * COSZ**1.5* (1.0 + 1.5*CLD**1.5)
      ZM10    = ZM1
      IF( ZM10.GT.ZMSTRO ) ZM10= ZMSTRO
      ZM1    = ZM10 + DM1
      IF( ZM1.GT.ZMSTR ) ZMSTR = ZM1
      ZM2    = ZM1 + TOC + DM1*0.5
      ZM30     = 1.0 + TOC
      ZM3      = ZM30
      IF( ZM2.GT.1.0 ) ZM3 = ZM2 + TOC
C
      SIG1=0.0
      SIG2=0.0
      IF( ZM2.NE.ZM1)  SIG1 = 0.5*((SPEED -ZM1)/(ZM2 -ZM1))
      IF( ZM3.NE.ZM2)  SIG2 = 0.5*(1.0 + (SPEED -ZM2)/(ZM3 -ZM2))
C
      TOCL    = 1.0/(4.4*ARWS*COSZ**1.5    )
      DTOCL   =(TOC - TOCL)/COSZ
      IF( DTOCL.LT.0.0) DTOCL = 0.0
      ARDT   = ARWS * DTOCL
      IF( DTOCL.GT.0.07) DTOCL = 0.07
      IF( ARDT.GT.0.1 ) ARDT = 0.1
      GAMMA = 9.0 * (DTOCL/(1.0 + 0.5 * ARDT) )
C
      GAMAO   = GAMMA
      IF( ZM30.NE.ZM10 ) GAMMA = GAMAO *(ZM3-ZM1)/(ZM30-ZM10)
      IF (GAMMA.GT.GAMAO ) GAMMA = GAMAO
C
      XF      = (16.0 +3.0*ARWS**2)/(8.0 +5.0*ARWS**2)
      XKB     =(1.0 +DOB)*(1.0 -DOB)**XF
      XKT    = 1.0
      IF(SPEED.GE.ZM1.AND.SPEED.LE.ZM2) XKT = 1.0 -(4.0*SIG1*(1.0
     1                     - SIG1))**3 * GAMMA
      IF(SPEED.GT.ZM2.AND.SPEED.LE.ZM3) XKT = 1.0 -(4.0*SIG2*(1.0
     1                     - SIG2))**3 * GAMMA
    5 CONTINUE
```

```
C
      Y1     = 2.0 + 0.66667 * SQRT(TAPER) - TAPER**2
      Y      = (1.0 + PI *ARWS)/(3.0 + PI * ARWS)   * Y1
      TWOS    = 1.0
      IF( ZMSTR.LT.1.0 ) TWOS = 1. + EPSL/SQRT(1.-ZMSTR**2)
C
      CLAO    = (TWOS * PI * ARWS)/(TWOS + SQRT(TWOS + (1.0 -
     1           COSZ**1.3334) * (ARWS/(2.0*COSZ))**2 ) )
      IF( SPEED.GT.0.0 ) BETAP = (SPEED - ZMSTR) * (1.0 +
     1                 (ZMSTR/SPEED)**Y )**2
      Z1     = PI * ARWS / CLAO
      Z1     = 3.0 * Z1 * ( Z1 - 1.0 ) * COSZ**0.6667
      Z      = ZMSTR * CLAO  + ARWS**2/Z1
C
C
      IF(SPEED.GT.ZMSTR)   GO TO 10
C
      CLAB    = (.0548311*TWOD*ARWS)/(TWOD + SQRT(TWOD + (1.0 -
     1           COSZ**1.3334 *(SPEED/ZMSTR)**2.667)*(ARWS/(2.*COSZ))**2
     2                                                               ))
      GO TO 20
   10 CLAB  = 1.0/(57.3 *((ZMSTR/SPEED)**Z /CLAO  + BETAP/4.0) )
C
C
C
C
   20 CONTINUE
      CLA     = CLAB * XKT * XKB * SPLAN/SREF
C
C
C
   25 IF( KPRINT(11).EQ.0 ) GO TO 30
      WRITE(6,1000) SPEED, CLA, CLAB, XKT, XKB, SPLAN, SREF,
     1               ZMSTR, ZM1, ZM2, ZM3, COSZ, CSUBO, ARDT,
     2               TOC, TAPER, ARWS, DMSTR, CLD, DM1, DOB, TWOD, SWPMC
 1000 FORMAT(10X*AER2 DUMP*, 6F15.5 /(19X,6F15.5) )
C
   30 RETURN
      END
```

CC = 00095

```
      SUBROUTINE TAIL(SPEED)
C
C     COMPUTES TAIL FLOW FIELD AND LIFT
C
      COMMON /BLKCLA/ CLAI(11)
C
      COMMON /BLKA02/ SREF, A1(428), HTLE, HTY, HTZ, TINC
      COMMON /BLKBD1/ XSWPL(11), YRMIN(11)
C
      COMMON /BLKG01/ G1(79), AR, TR, SPLAN, CB, YB, XMAC, G2(16),
     1                SWPQC, G3(4), XHT, XCRTE, G4(92)
C
      COMMON /BLKC01/ C1(37), CDWING(4), C2(12), CLAW, CLAB, CLAT, A, B,
     1                AOH, C, ASTR, CLDH, C3(31), DEDA, C4(6)
C
      COMMON /BLKPRT/ KPRINT(50)
C
C
C  ** DOWNWASH IS CALCULATED ***
C
      ARHT   = G1(11)
      SEXHT  = G1(21)
      SWLHT  = A1(389)
      HTLAM  = A1(409)
      CAMHT  = A1(339)
      HTOC   = A1(349)
      SWMCHT = ATAN(TAN(SWLHT) -2./ARHT * (1.-HTLAM)/(1.+HTLAM))
C
   10 ZKA    = 1./AR -1./(1. + AR**1.7)
      IF( AR.LT.2.3 ) ZKA = 0.37 - 0.0567 * AR
      ZKTR   = (10. - 3.0 * TR)/7.0
      B      = SQRT( AR * SPLAN )
C
      DEDA   = 0.0
      XLHT   = XHT - XMAC
      IF( XLHT.LE.0.0 ) GO TO 30
C
      HHT    = HTZ
C
      ZKH    = (1. - HHT/B)/(2.0 * XLHT/B)**0.33333
C
      COSQ   = SQRT(COS(SWPQC))
   20 DEDAO  = 4.44 * (ZKA * ZKTR * ZKH * COSQ )**1.19
C
      IF( KPRINT(21).EQ.0 ) GO TO 21
      WRITE(6,1000)CLAW,DEDAO,CDWING(1),CDWING(2),CDWING(3),CDWING(4),
     1            HTLE,A1(281),CB,ZKA,ZKTR,ZKH,COSQ
   21 CONTINUE
      CALL AER2(0.1, CLAMO)
      DEDA   = DEDAO *CLAW/CLAMO
C
C  ** DYNAMIC PRESSURE AT THE TAIL  ***
C
   30 CDO    = CDWING(1)+CDWING(2)+CDWING(3)+CDWING(4)
C
      XOC    = (XHT - XCRTE)/CB
```

```
      DQQQ   = 0.0
      IF( XOC.LE.0.0 ) GO TO 50
C
      ZWOC   = 0.68 * SQRT(CDO * (XOC + 0.15))
      DQQQO  = 2.42 * SQRT(CDO)/(XOC + 0.3)
C
      GAMMA  = ATAN(HTZ/(XOC*CB))
      AW     = 2.0
      CL     = CLAW* AW
      EW     = 0.51566 * CL/AR
      ZOC    = XOC * TAN(GAMMA + EW - AW/57.3)
      ZOZW   = ZOC/ZWOC
C
      IF( ZOZW.GE.1.0 ) GO TO 50
C
      DQQQ   = DQQQO * (COS(1.570796* ZOZW))**2
C
   50 QQQ    = 1.0 - DQQQ
C
C   ** CARRY-OVER FACTORS ZKWB AND ZKBW ARE COMPUTED ***
C

   60 DOBHT  = 2.0 * HTY/(2.0 * HTY + SQRT(ARHT * SEXHT) )
C
      ZKWB   = 1.0028 +.7116*DOBHT +.42*DOBHT**2 -.1366*DOBHT**3
      ZKBW   = .0004 +1.2662*DOBHT +.6018*DOBHT**2 +.1263*DOBHT**3
C
C   ** LIFT-CURVE-SLOPE OF THE EXPOSED SURFACE IS COMPUTED ***
C
      CLAI(1)  = SEXHT
      CLAI(2)  = HTOC
      CLAI(3)  = HTLAM
      CLAI(4)  = ARHT
      CLAI(5)  = 0.0
      CLAI(6)  = CAMHT
C
      CLAI(7)  = 0.0
      CLAI(8)  = 0.0
      CLAI(9)  = 0.0
      CLAI(10) = SWMCHT
C
   70 CALL AER2(SPEED, CLA)
      CLAT   = CLA * (ZKWB + ZKBW) * (1. - DEDA) * QQQ
      CLDH   = CLA * ZKWB * QQQ
C
C   ** INDUCED DRAG FACTOR FOR TAIL IS COMPUTED ***
C
      CALL LNTP(SWLHT, RTSUB, XSWPL, YRMIN, 11, 2)
      RTAIL  = RTSUB
      IF( SPEED.LE.0.9 ) GO TO 80
      ZNDM   = 12.0 * (COS(SWLHT)**1.6) * (SPEED - 0.9)
      FNDM   = 1.0/(1. + ZNDM + ZNDM*ZNDM)
      RTAIL  = RTSUB * FNDM
C
   80 HTK1   = (1.-RTAIL) * 0.01745 /(CLA * ZKWB)
      HTK2   = RTAIL * 0.31831/ARHT * SREF/SEXHT
```

```
      HTK     = HTK1 + HTK2
C
      A       = HTK * CLDH * CLDH
      B       = 2.0 * HTK * CLAT * CLDH
      AOH     = (1.-DEDA) * (A1(279) - TINC)
C
C
      IF( KPRINT(21).EQ.1 ) WRITE(6,1000) CLAT, SPEED, CLA, ZKWB, DGBHT,
     1                       ZKBW, DEDA, QOQ, CLDH, A, B, AOH,DEDAO,
     2                 RTSUB,RTAIL,HTK1,HTK2,HTK
C
      RETURN
 1000 FORMAT(10X,*TAIL LIFT DUMP* /(1X, 7F15.5) )
      END
```

CC = 00126

```
      SUBROUTINE AALO(SPEED)
C
C     CALCULATE ZERO LIFT ANGLE OF ATTACK
C
      COMMON /BLKA02/ A1(433)
C
      COMMON /BLKG01/ G1(44), DOB, TOC, CLD, G2(33), TAP, G3(20),
     1                    ESWQC, G4(98)
C
      COMMON /BLKC01/ C4(16), CLA, ALO, C5(35), C1, C2, C3, C6(44)
C
      COMMON /BLKPRT/ KPRINT(50)
C
C
C
      ALOC    = 0.0
      ALOT    = 0.0
      ALOI    = 0.0
      TWIST   = G4(10)
      RINC    = A1(280)
      DINC    = G4(11)
      TINC    = A1(433)
C
      XMCR    = 0.75
      XMNO    = SPEED * COS(ESWQC)
C
      TOCO    = TOC/COS(ESWQC)
      IF( TOCO.LT.0.1 ) XMCR = 0.75 + 1.25 * (0.1 - TOCO)
      DALODC = 5.6
      IF(  XMNO.GT.XMCR ) DALODC = 5.6 - 249.0 * (XMNO - XMCR)**2
      IF( DALODC.LT.0.0 ) DALODC = 0.0
C
      ALOC    = - DALODC * CLD
C
C
  200 IF( TWIST.EQ.0.0 ) GO TO 300
      IF( SPEED.GT.1.0 ) GO TO 300
      TAU = TWIST
      BETA = SQRT(1.0 - SPEED**2)
      SWPQCB = 90.0
      IF( BETA.GT.0.0 ) SWPQCB = ATAN(TAN(ESWQC)/BETA) * 57.29578
      DALO    = 0.093 - 0.000571*SWPQCB + 0.5761*TAP -0.2645*TAP**2
      ALOT    = - DALO * TAU
C
  300 IF( DINC.EQ.0.0.AND.TINC.EQ.0.0 ) GO TO 400
      ZKBW    = .0004 +1.2662*DOB +.6018*DOB**2 +.1263*DOB**3
      ZKWB    = 1.0028 +.7116*DOB +.42 *DOB**2 -.1366*DOB**3
      CLAB    = C3 + ZKBW/(ZKBW + ZKWB) * C1
      ALOI    = (CLAB * DINC + C2 *(DINC - TINC))/CLA
     1                + (RINC - DINC)
C
  400 ALO     =          + ALOC + ALOT + ALOI
      IF( KPRINT(13).GT.0 ) WRITE(6,1000) ALO, ALOC, ALOT,
     1                    ALOI, SPEED, ESWQC, TOC, CLD, TAU, SWPQCB, DOB,
     2                      TAP, C1, C2, C3, CLAB, ZKBW, ZKWB, DINC,TINC
     3                    ,RINC
```

```
C                                                                        R1T2979
 1000 FORMAT(5X,*AALO DUMP*,5X,*ALO =*,F6.3,5X,*ALOC =*,F6.3,            R1T2980
     1        5X,*ALOT =*,F6.3,5X,*ALOI =*,F6.3 /(5X,7F15.5) )           R1T2981
      RETURN                                                             R1T2982
      END                                                                R1T2983
                                                              CC = 00061
```

```
      SUBROUTINE CLBRK(SPEED, RE, RNOFT)
C
C     CALCULATES LIFT BREAK CL AND CLMAX
C
      COMMON /BLKG01/ G1(45), TOC, CLD, SEXW, G2(31), AR, TR, SPLAN,
     1                CB, G3(18), SWPQC, SWPLE, G4(97)
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKA02/ SREF, A1(432)
C
      COMMON /BLKPRT/ KPRINT(50)
      COMMON /BLKC01/ C3(16), CLA, ALO, C4(35), CLAW, CLAB, CLAT, C6(6),
     1                CLPB, CLDB, CLMAX, ABRK, AMAX, DAMAX, DEL,
     2                CLS, ARLO, C5(29)
C
      COMMON /BLKMAP/ MAP, TRANS, DY, AMAP(22), BMAP(11)
      COMMON /BLKCLB/ X1(6), Y1(6), X9(8), Y9(8), XAR(5), YDCL(5)
C
      COMMON /BLKBD2/ AA(22), BB(22), CC(22), DD(22), XT(22)
C
      COMMON /BLKBD5/ XDYO(6), YXMN(6), Z1BD5(6,6)
C
      COMMON /BLKMAX/ XTR(6), YC1(6), YC2(6), XSWP(4), YA(4),
     1                YB(4), XDY(8), XM(4), CTAB(8,4), DTAB(8,4),
     2                XXCLM(13), YYDY(6), FCLMX(13,6), XXC2(9),
     3                YYMACH(5), FDCLMX(9,5),XDY1(9), YXMT(4),
     4                 ZC1MAX(9,4), XDY2(8), YFOC(6), ZDC1M(8,6),
     5                Z2DC1M(8,6)
      COMMON /BLKMX2/  XSP(8), YDY(6), FDA(8,6), XAB(6), YCO(8),
     1                FKVOFM(6,8), XANG(10), YRTOC(7), FRA(10,7)
      COMMON /BLKMX4/ XC2(6),YAST(9),FDAM(6,9),XCT(6),YM(5),FDAM2(6,5)
      COMMON /BLKMX5/ XXXCLM(9), FCLMXX(9,6)
C
      DIMENSION XF1(7), YF1(7), YF2(7)
C
      DATA   XF1 / 5.0, 6.0, 6.47712, 6.77815, 6.95424, 7.39794, 8. /,
     1       YF1 / -.125, -.11, -.02, -.01, 0., 0.03, 0.035 /,
     2       YF2 / -.1063, -.0713, -.055, -.0175, 0., 0.0375, 0.0463/
C
      IF( SPEED. GE.1.0 ) GO TO 20
C
      NI     = A1(234)
      CONCL  = A1(276)
      DYO    = AMAP(NI) * TOC
      IF( DYO.LT.0.8 ) DYO = 0.8
      IF( DYO.GT.2.4 ) DYO = 2.4
C
      XMN     = SPEED * COS(SWPQC)
      IF( XMN.LT.0.2 ) XMN = 0.2
      XMNO    = XMN
      CLIN    = CLD/COS(SWPQC)**2
      IF( XMN.GT.0.7 ) XMNO = 0.7
C
      ABRKO   = DLNT(DYO, XMNO,XDYO, YXMN, Z1BD5, 6,6, 6, 4,2)
      ABRK    = ( ABRKO + (12.05-4.1*XMN)*CLIN ) * COS(SWPQC)
```

```
      ABRKLO = 2./COS(SWPLE)
      IF( ABRK.LT.ABRKLO ) ABRK = ABRKLO
      CLPBO  = CLA * ABRK
C
      DCLPB  = 0.0
      FPB    = 1.54
      IF( XMN.GT.0.5 ) FPB = 1.54 - 2.9 * (XMN - 0.5)
      IF( CONCL.GT.0.0 ) DCLPB = FPB * ((CONCL + .0643)*COS(SWPQC))**2
C
      CLPB   = CLPBO + DCLPB
C
      DCLDRE = 0.1226 -.00714*SPEED -.12857*SPEED**2
C
      IF( NI.LE.9 ) DY = AMAP(NI)*TOC +BMAP(NI)*CLD
      IF( NI.GT.9 ) DY = AMAP(NI)*TOC +1.75*CLD
C
      TRANS  = 0.0
      IF( DY.GT.1.65 ) TRANS = (DY - 1.65)/.4
      IF( DY.GE.2.05 ) TRANS = 1.0
      IF( SWPLE.GE.0.87 ) TRANS = 0.0
      CLDB   = CLPB + TRANS *(-.0376 -.24414*SPEED -.0685*SPEED**2
     1                +.4149*SPEED**3 + RE * DCLDRE )
C
      GO TO 30
C
   20 BETAC  = SQRT(SPEED * SPEED - 1.)/TAN(SWPLE)
      CLSB1  = 0.85
      CLSB9  = 0.85
      IF( SWPLE.GT.0.7243 ) CALL LNTP(SWPLE, CLSB1, X1, Y1, 6, 4)
      IF( SWPLE.GT.0.3665 ) CALL LNTP(SWPLE, CLSB9, X9, Y9, 8, 4)
      CALL LNTP(AR, DCLAR, XAR, YDCL, 5, 4)
      CLSB9  = CLSB9 + DCLAR
      CLSB   = CLSB1 + (CLSB9 - CLSB1) *(BETAC - .1) * 1.25
      CLSB   = CLSB + 0.5 * CLD
      CLPB   = CLSB
      CLDB   = CLSB
C
C     SUPERSONIC MAXIMUM LIFT COEFFICIENT     ****
      CNA04B = CLAW * 14.325 * SQRT(SPEED*SPEED -1.)
C
      OM     = 1./SPEED
      CLMAX  = 0.7722 + 0.3384 * OM +1.1648 * OM*OM -0.8215 * OM**3
C
      CM1    = 1. - CNA04B
      IF(CM1.LT.0.0 ) CM1 = 0.0
C
      CLMAX  = CLMAX - 0.048 * CM1
C
      AMAX   = 68.5429 -177.2327 * OM +461.9204 *OM*OM
     1                 -629.4522 * OM**3 +321.4001 * CM**4
C
      AMAX   = AMAX +15.8074 -3.0001*CNA04B -12.8073*CNAC48**2
C
      IF( CLMAX.GT.1.25 ) CLMAX = 1.25
      IF( AMAX.GT.54.5  ) AMAX  = 54.5
C
```

```
      GO TO 300
   30 CONTINUE
C
      CLMAX  = 0.0
      NI     = A1(234)
      DY     = AMAP(NI) * TOC
      CALL LNTP(TR, C1, XTR, YC1, 6, 4)
      CALL LNTP(TR, C2, XTR, YC2, 6, 4)
C
      C1COS  = (C1 + 1.) * COS(SWPLE)
      ARUP   = 4./C1COS
      ARLO   = 3./C1COS
C
C
      XMT      = XT(NI)
      IF( XMT.LT.0.3 ) XMT = 0.3
      IF( XMT.GT.0.45) XMT = 0.45
      IF( AR.LE.ARLO ) GO TO 200
C
      C1MAXB = DLNT(DY, XMT, XDY1, YXMT, ZC1MAX, 9, 4, 9, 2,2)
      FOC    = 5.500 * CLD
      DC1MAX = DLNT(DY, FOC, XDY2, YFOC, ZDC1M, 8, 6, 8, 2,2)
      IF( XMT.GE.0.35 ) DC1MAX = DLNT(DY,FOC,XDY2,YFOC,Z2DC1M,8,6,8,2,2)
C
      RNCB    = ALOG10(RNOFT * CB)
      CALL LNTP(RNCB, F1, XF1, YF  , 7, 2)
      CALL LNTP(RNCB, F2, XF1, YF  , 7, 2)
      DC1RN   = F1 + F2 * DY
C
      C1MAX   = C1MAXB + DC1MAX + DC1RN
C
      SWEEP = SWPLE * 57.2956
  100 CALL LNTP(SWEEP, A, XSWP, YA, 4, 4)
      CALL LNTP(SWEEP, B, XSWP, YB, 4, 4)
C
      DYM14   = DY - 1.4
      IF( DYM14.LT.0.0 ) DYM14 = 0.0
      IF( DY.GT.2.5    ) DYM14 = 1.1
C
      CLMOC1 = A - B * DYM14
      CLMAX  = CLMOC1 * C1MAX
C
      C       = DLNT(DY, SPEED, XDY, XM, CTAB, 8, 4, 8, 2,2)
      D       = DLNT(DY, SPEED, XDY, XM, DTAB, 8, 4, 8, 2,2)
C
      DCLMAX = C   + (D - C) * SWEEP/60.
      CLMAX  = CLMAX + DCLMAX
C
  120 DAMAX  = DLNT(SWEEP, DY, XSP, YDY, FDA, 8, 6, 8, 2,2)
      AMAX   = CLMAX/CLAW  +ALO  +DAMAX
      CLS    = CLA * (AMAX - 2.*DAMAX - ALO)
      GO TO 300
C
C     LOW ASPECT RATIO METHOD
C
  200 CONTINUE
```

```
C                                                                       R1T3153
      ABETA  = AR/SQRT(1. - SPEED * SPEED)                              R1T3154
      XCLM   = ABETA * C1COS                                            R1T3155
      IF( XMT.LE.0.35.OR.XCLM.GE.2.0 ) CLMXB =  DLNT(XCLM, DY, XXCLM,   R1T3156
     1                      YYDY, FCLMX, 13, 6, 13, 2,2)                R1T3157
      IF( XMT.GT.0.35.AND.XCLM.LT.2.0 ) CLMXB =  DLNT(XCLM, DY, XXXCLM, R1T3158
     1                      YYDY, FCLMXX, 9, 6, 9, 2,2)                 R1T3159
C                                                                       R1T3160
      C2TAN  = (C2 + 1.) * AR * TAN(SWPLE)                              R1T3161
      DCLMX  = DLNT(C2TAN, SPEED, XXC2, YYMACH, FDCLMX, 9, 5, 9, 2,2)   R1T3162
C                                                                       R1T3163
      CLMAX  = CLMXB + DCLMX                                            R1T3164
      AST    = AR * COS(SWPLE) * (1.+ 4.0 * TR*TR)                      R1T3165
      AMAXB  = 35.0                                                     R1T3166
      IF( XCLM.GT.0.9 ) AMAXB = 49.8473 -20.6922*XCLM +5.0674*XCLM**2   R1T3167
     1                           -0.4279*XCLM**3                        R1T3168
      IF( XCLM.GT.3.6 ) AMAXB = 21.0                                    R1T3169
      IF( C2TAN.LE.4.5 ) DAM = DLNT(C2TAN,AST, XC2, YAST,FDAM,6,9,6,4,2)R1T3170
      IF( C2TAN.GT.4.5 ) DAM = DLNT(C2TAN,SPEED,XCT,YM,FDAM2,6,5,6,4,2) R1T3171
C                                                                       R1T3172
      AMAX   = AMAXB + DAM                                              R1T3173
C                                                                       R1T3174
  300 CONTINUE                                                          R1T3175
C     TAIL CONTRIBUTION TO CLMAX        ******                          R1T3176
C                                                                       R1T3177
      DEL    = CLAT * 57.3 * SIN(AMAX/57.3) * COS(AMAX/57.3)**2         R1T3178
      CLMAX  = CLMAX + DEL                                              R1T3179
C                                                                       R1T3180
      IF( KPRINT(16).EQ.1 ) WRITE(6,1000) CLPB, CLDB, ABRKO, ABRK,      R1T3181
     1              XMN, DYO, DCLPB, TRANS, BETAC, DY, C1, C2, AMAX, ARUP,R1T3182
     2                  ARLO,XMT,C1MAXB,DC1MAX,CLMOC1,DCLMAX,DEL,       R1T3183
     3              XCLM, CLMXB, C2TAN, DCLMX, CLMAX, DC1RN, RNCB       R1T3184
 1000 FORMAT(10X,*CLBRK DUMP* /(1X, 7F15.5) )                           R1T3185
C                                                                       R1T3186
      RETURN                                                            R1T3187
      END                                                               R1T3188

                                                      CC = 00204
```

```
      SUBROUTINE CDL1(SPEED, RNOFT, FK, DELCL, PRIMEK, AKD, AKB)
C
C     COMPUTES DRAG DUE TO LIFT CONSTANTS
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKA02/ SREF, A1(223), CBAR(10), TW(10), CAM(10), TOC(10),
     1                A2(12), CONCL, A3(25), YW(11), A4(120)
C
      COMMON /BLKG01/ G1(44), DOB, TOCW, CLD,
     1                G3(12), XLESW(10), G4(10), AR, TR, SPLAN,CB,
     2                G5(18), SWPQC, SWPLE, G6(11), CBAR2, CLDS, TOCS,
     3                G7(83)
C
      COMMON /BLKBD1/ XSWPL(11), YRMIN(11)
C
      COMMON /BLKBD2 / AA(22), BB(22), CC(22), DD(22), XT(22)
      COMMON /BLKA16/ XSWP(7), YTR(4), FEP35(7,4), FEP7(7,4),
     1                XCLDB(7), YAKB(7)
C
      COMMON /BLKC01/ C1(100)
C
      COMMON /BLKPRT/ KPRINT(50)
C
      DIMENSION OCURV(13), RCURV(13), XRN(9), YDRT1(9), YDRT2(9)
C
      DATA OCURV / 2., 4., 6., 10., 20., 30., 40., 50., 60.,
     1             80., 100., 130., 200./,
     2     RCURV / 0.17,0.37,0.48,0.6,0.73,0.782,0.81,0.83,0.84,
     3             0.855,0.865,0.874,0.874 /
      DATA  XRN   / 0., 5., 7., 10., 20., 40., 100., 300., 600. /,
     1     YDRT1 / .07,.07,.05,.025, 5*0.0 /,
     2     YDRT2 / .175,.175,.165,.12,.05,.03,.015,.005,0.0 /
C
C
      CLM     = C1(17) * 57.29578 * SREF/SPLAN
      BO2X    = YW(1+NPNLS) - YW(1)
      FKL1    = 0.0
      DCL1    = 0.0
      FKL2    = 0.0
      DCL2    = 0.0
      CLMO    = CLM
      CLM2    = C1(77) * 57.29578 * SREF/SPLAN
      RBAR    = 0.0
C
      CLDB    = C1(64)
      CALL LNTP(CLDB, AKB, XCLDB, YAKB, 7, 2)
      FAKB    = 1.24 -.04 * RNOFT * CB * 1.0E-6
      IF( FAKB.LT.1.0 ) FAKB = 1.0
      AKB     = FAKB * AKB * SREF/SPLAN
C
      B       = -1.41 +1.442*TR -1.26*TR**2 +.528*TR**3
      C       = 0.7125 -1.497*TR +1.476*TR**2 -.6909*TR**3
      EO      = 1.0 + B * DOB + C * DOB**2
C
      FMACH   = SPEED
```

```
      FMCRO   = C1(72)
      FML1    = C1(73)
      FML2    = C1(74)
      IF( FMACH.GT.FMCRO ) FMACH = FMCRO
      IF( SPEED.GE.FMCRO.AND.SPEED.LT.FML2 ) CLMO= C1(76) *
     1           57.29578 * SREF/SPLAN
      IF( SPEED.GT.1.0 ) CALL ADCL(SPEED, CLOPT)
      IF( SPEED.LT.FML2 ) CALL ADCL(FMACH, CLOPT)
C
      NI      = TW(1)
      DO 100 I = 1, NPNLS
C
      TC      = TOC(I)
      RLEOC   = AA(NI) +BB(NI)*TOC(I) +CC(NI)*TOC(I)**2
     1            +DD(NI)*TOC(I)**3
      IF( NI.EQ.8 ) RLEOC = 0.88216 * TOC(I)**1.606
      RLE     = RLEOC * CBAR(I)
      IF( ISWP.EQ.0.OR.I.NE.NPNLS ) GO TO 90
      RLEOC   = AA(NI) +BB(NI)*TOCS + CC(NI)*TOCS**2 + DD(NI)*TOCS**3
      IF( NI.EQ.8 ) RLEOC = 0.88216 * TOCS**1.606
      RLE     = RLEOC * CBAR2
      TC      = TOCS
C
   90 CONTINUE
C
      RNLER   = RLE * RNOFT/10.0**3
      COTANS = 5.0 - 6.511 * XLESW(I)
      IF( SWPLE.GT.0.35 ) COTANS = 1./TAN(XLESW(I))
C
      OMEGA   = RNLER * COTANS * SQRT(1. - (FMACH * COS(XLESW(I)))**2 )
      CALL LNTP(OMEGA, RT, OCURV, RCURV, 13, 2)
      CALL LNTP(XLESW(I),RMIN,XSWPL,YRMIN, 11, 2)
      IF(RT.LT.RMIN )  RT = RMIN
      DRT     = 0.0
C
      CALL LNTP(RNLER, DRT1, XRN, YDRT1,9,2)
      CALL LNTP(RNLER, DRT2, XRN, YDRT2,9,2)
      IF( TC.GT.0.03 ) DRT = DRT1 * (TC - 0.03)/0.03
      IF( TC.GT.0.06 ) DRT = DRT1 + DRT2 * (TC - 0.06)/0.06
      RT      = RT + DRT
C
      RI      = RT +(0.824 -RT) *(CLD + CONCL)/0.6
C
      IF(RI.GT.0.874 ) RI = 0.874
C
      RBAR    = RBAR + RI * (YW(I+1) - YW(I))/BO2X
  100 CONTINUE
C
      AT      = AR * TR/COS(SWPLE)
      DELR    = 0.0482*AT -0.01102*AT**2 +0.001197*AT**3
     1           -0.00004833*AT**4
C
      R       = RBAR + DELR
C
      FK      = ((1.-R)/CLMO +R/(3.14159*AR*EO)) * SREF/SPLAN
C
```

```
      E      = 1./(3.14159 * AR * FK) * SREF/SPLAN
      DELCL  = CLOPT * (1.-E)
      PRIMFK = 0.518/SQRT(AR)
C
      SWPC4  = SWPQC * 57.29578
      IF( SWPC4.GT.60.0 ) SWPC4 = 60.0
      EP35   = DLNT(SWPC4, TR, XSWP, YTR, FEP35, 7, 4, 7, 2,2)
      EP7    = DLNT(SWPC4, TR, XSWP, YTR, FEP7,  7, 4, 7, 2,2)
      IF( AR.LE.3.5 ) EP = 1. - (1.-EP35) * AR/3.5
      IF( AR.GT.3.5 ) EP = EP35 + (EP7 - EP35)*(AR-3.5)/3.5
C
      EPP    = EP * (1.-DOB**2)
      AKD    = 1./(3.14159 * AR * EPP) * SREF/SPLAN
C
C
      RO     = R
      H      = 1.1
      ARTANS = AR * TAN(SWPLE)
      IF( ARTANS.GT.3.5 ) H = 1.1 + 0.1*(ARTANS - 3.5)
      IF( SPEED.GE.1.0 ) PRIMEK = H/CLM * SREF/SPLAN
C
      IF( SPEED.LE.FML2 )  GO TO 200
      ZNDM   = 12. * (COS(SWPLE)**1.6) * (SPEED-FMCRO)
      FNDM   = 1./(1. + ZNDM + ZNDM**2)
      R      = RO * FNDM
      FK     = ((1.-R)/CLM  +R/(3.14159*AR*EO)) * SREF/SPLAN
      DELCL  = CLOPT
C
      GO TO 500
C
  200 CALL KGIN(CLDB, FK, DELCL, SPEED, FKL1, DCL1)
      IF( SPEED.LE.FML1 ) GO TO 500
C     DRAG POLAR IS CALCULATED BY LINEAR INTERPOLATION BETWEEN THE
C     LIMITS FML1 AND FML2.
C
      ZNDM   = 12. * (COS(SWPLE)**1.6) * (FML2 - FMCRO)
      FNDM   = 1./(1. + ZNDM + ZNDM**2)
      RL2    = RO * FNDM
      FKL2   = ((1.-RL2)/CLM2 + RL2/(3.14159*AR*EO)) * SREF/SPLAN
      CALL ADCL(FML2, DCL2)
C
      CALL KGIN(0.50, FK, DELCL, FML1, FKL1, DCL1)
C
      FK     = FKL1 + (FKL2-FKL1)*(SPEED-FML1)/(FML2-FML1)
      DELCL  = DCL1 + (DCL2-DCL1)*(SPEED-FML1)/(FML2-FML1)
C
  500 IF( KPRINT(22).EQ.0 ) RETURN
      WRITE(6,1000) SPEED, FMACH, FK, DELCL, PRIMEK, AKD, AKB,
     1              CLDB, CLM, SREF, SPLAN, FMCRO, FML1, FML2,
     2              RLE,RNLER,OMEGA,R,RBAR,RT,CLD,CONCL,E,AR,SWPC4,
     3              EP, DOB, SWPLE, FKL1, DCL1, FKL2, DCL2,RMIN
     4                , (TOC(I), CBAR(I), XLESW(I), YW(I), I = 1,NPNLS),
     5                YW(1+NPNLS)
C
 1000 FORMAT(5X, *CDL1 DUMP* /(1X, 7F15.5) )
      RETURN
```

END

RIT3358

CC = 00169

```
      SUBROUTINE ADCL(SPEED, CLOPT)
C
C     POLAR AXIS DISPLACEMENT
C
      COMMON /BLKA02/ SREF, A1(432)
C
      COMMON /BLKG01/ G1(46), CLD, SEXW, G2(33), SPLAN, G3(20),
     1                ESWLE, G4(97)
C
      CONCL  = A1(276)
C
      CLOPT  = 0.0
      IF (SPEED .GE. 1.0) GO TO 10
      FOC    = 0.06 * CLD
C
      IF( CLD.GT.0.0 ) CLOPT = -.001 +16.934*FOC -216.2697*FOC**2
     1                         +1781.3562*FOC**3
      IF( CONCL.GT.0.0)CLOPT = CLOPT -.0017 +1.1334*CONCL
     1                      -1.8498*CONCL**2 +1.0605*CONCL**3
      IF( A1(234).EQ.8. ) CLOPT = 0.51951 * CLD**0.75
      GO TO 20
C
   10 BETAT  = 10.
      IF( ESWLE.GT.0.0 ) BETAT = SQRT(SPEED**2 -1.)/TAN(ESWLE)
      DELCL  = CLD * (0.25 - 0.225 * BETAT )
      IF( BETAT.GE.1.11 ) DELCL = 0.0
      CLOPT  = DELCL
C
   20 CLOPT  = CLOPT * SPLAN/SREF
      RETURN
      END
```

CC = 00031

```
      SUBROUTINE KGIN(CLDB, AKIN,DECLIN,SPEED, AKOUT,DCLOUT)
C
C     COMPUTES POLAR USING LEAST-SQUARES CURVE FIT
C
      COMMON /BLKPRT/ KPRINT(50)
      DIMENSION CL(11), CDL(11), SA(12), RE(11)
C
      DCL     = CLDB/10.0
      CL(1) = 0.0
C
      DO 20 I = 1, 11
      IF( I.NE.1 ) CL(I) = CL(I-1) + DCL
      CLX     = CL(I)
C
      CALL CDDR1(CLX, SPEED, CDR)
C
      IF( I.EQ.1 ) CDRO = CDR
      DCDR   = CDR - CDRO
C
   40 CDL(I) = AKIN * (CL(I) - DECLIN)**2 + DCDR
   20 CONTINUE
C
      CALL PLSQ(CL,CDL,7,2,SA,0,EMAX,ERMS,EMEQ)
C
      AKOUT    = SA(1)
      DCLOUT   = -SA(2)/(2.0 * AKOUT )
      ERR      = SA(3) - AKOUT * DCLOUT**2
C
      IF( ABS(ERR).LT.0.001.OR.KPRINT(14).EQ.0 ) GO TO 30
C
      WRITE (6,1000) SPEED,ERR,AKOUT,DCLOUT,
     1         AKIN, DECLIN, ( CL(I), CDL(I), I = 1,7 )
 1000 FORMAT(10X *KGIN* 5X,*MACH =*F8.5,2X*ERR =*F8.5,2X,4F8.5,/
     1         1X, 4F8.5      )
C
   30 RETURN
      END
```

CC = 00037

```
      SUBROUTINE CDL2(SPEED, CL, AEROK, DELCL, PRIMEK, AKD, AKB, CDL)
C
C     COMPUTES DRAG DUE TO LIFT
C
      COMMON /BLKPRT/ KPRINT(50)
C
      COMMON /BLKC01/ C1(100)
C
      CLPB    = C1(63)
      CLDB    = C1(64)
C
C
   20 CONTINUE
      CDL    = AEROK * (CL - DELCL)**2
      IF( CL.LE.CLPB ) GO TO 500
C
C     DRAG DUE TO LIFT ABOVE POLAR BREAK
      CDL    = CDL + PRIMEK * (CL - CLPB)**2
      IF( CL.LE.CLDB ) GO TO 500
C
C     DRAG DUE TO LIFT ABOVE DRAG BREAK (CLDB)
C
      IF( SPEED.GE.1.0 ) GO TO 410
      CDPCB   = AEROK * (CLDB - DELCL)**2  - AKD * CLDB**2
     1              + PRIMEK * (CLDB - CLPB)**2
C
      DCDB    = AKB * (CL - CLDB)**2
      DCDB    = DCDB + 0.08 * SQRT(DCDB)
C
      CDL     = CDPDB + DCDB + AKD * CL**2
C
  400 IF( SPEED.LT.1.0 ) GO TO 500
  410 CONTINUE
      CDL     = (AEROK - PRIMEK) * (CLPB - DELCL)**2
     1          + PRIMEK * (CL - DELCL)**2
C
C
C
  500 CONTINUE
      IF( KPRINT(14).LE.0 ) GO TO 50
      WRITE(6,1000) SPEED, CL, CDL, AEROK, DELCL, PRIMEK, CLPB,
     1              CLDB, AKD, AKB
      KPRINT(14) = KPRINT(14) - 1
   50 CONTINUE
      RETURN
 1000 FORMAT (10X, *CDL2 DUMP*/(1X, 7F15.5 )  )
      END
```

CC = 00047

```
      SUBROUTINE AERA(SPEED, CL, ALPHA)
C
C     CALCULATES ANGLE OF ATTACK
C
      COMMON /BLKC01/ C1(100)
C
      COMMON /BLKG01/ G1(44), DOB, TOC, CLD, SEXW, SWPMC,
     1                G2(30), AR, TR, SPLAN, G3(6), YIX, YOX,
     2                SIX, SOX, ARI, G4(4), SWLEI, G5(3), SWPQC,
     3                  SWPLE, SWPTE, G6(96)
C
      COMMON /BLKA02/ SREF, A1(432)
C
      COMMON /BLKPRT/ KPRINT(50)
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKT22/ XIN13(10), YIN13(7), ZOUT13(10,7), ZOUT14(10,7)
      COMMON /BLKMX2/ XSWP(8), YDY(6), FDA(8,6), XAB(6), YCO(8),
     1                FKVOFM(6,8), XANG(10), YRTOC(7), FRA(10,7)
      COMMON /BLKMX6/ XX(12), XY(7), XF(12,7)
C
      DIMENSION CLTAB(13), ATAB(13)
      EQUIVALENCE (CLA[illegible](17)), (ALO,C1(18)), (CLAW,C1(54)),
     1            (CL[illegible](64)), (CLMAX,C1(65)), (AMAX,C1(67)),
     2            (DAMA[illegible]1(68)), (DEL,C1(69)), (CLS,C1(70)),
     3            (ARLO,C1(71)), (CDL,C1(6))
C
      SWEEP=SWPLF*57.29578
      KPRT   = KPRINT(17)
      CLVM1  = 0.0
C
   10 ALPHA = CL/CLA + ALO
      IF( SPEED.GT.1.0 ) RETURN
      IF( NPNLS.GT.1 ) GO TO 250
      IF( AR.LE.ARLO ) GO TO 100
C
   20 CONTINUE
C
      IF( KPRT.GT.0 ) WRITE(6,1000) CL, ALPHA, AR, ARLO, CLMAX, CLS,
     1                     DAMAX, CLA, ALO
C
      IF( CL.LE.CLS ) GO TO 200
C
C     HIGH ASPECT RATIO LIFT METHOD      ****
C
      DA     = 0.0
      DCL    = CLMAX - CLS
      DA     = ((CL - CLS)/DCL)**2 * DAMAX
   30 ALPHA = ALPHA + DA
      IF( CL.LE.CLMAX ) GO TO 200
      ALPHA = ALPHA + 5.0
      IF( ALPHA.GT.90.0 ) ALPHA = 90.0
C
      GO TO 200
  100 CONTINUE
```

```
C     LOW ASPECT  RATIO LIFT METHOD     *****
C
      AS      = 0.0
      Z       = 2. * COS(SWPMC) / AR
      HOB     = (AMAX - AS)/114.6 /AR    * 1.5 * (TR +TR**2)/(1.+TR+TR**2)
      XEP     = 1.0014 -1.969*HOB +3.0021*HOB**2 -2.0072*HOB**3
      DCLEP   = (Z + SQRT(1.+ Z**2))/(XEP*Z + SQRT(1. +(XEP*Z)**2))
      DCLPA   = (DCLEP -1.) * CLAW * 57.3
      CLPA    = CLA * 57.3
      DCLO    = - CLA * ALO
      CLVM1   = CLMAX - CLPA * SIN(AMAX/57.3) * COS(AMAX/57.3)**2 -DCLO
      CLVM2   = CLVM1 - DCLPA * SIN(AMAX/57.3) * COS(AMAX/57.3)**2
      IF( CLVM1.LE.0.0 ) DAMAX = AMAX - CLMAX/CLA - ALO
      IF( CLVM1.LE.0.0 ) CLS = CLA * (AMAX - 2.*DAMAX - ALO)
      IF( CLVM1.LE.0.0 ) GO TO 20
C
      CO      = TAN(SWPTE)/TAN(SWPLE)
      BETA    = SQRT(1. - SPEED**2)
      ABETA   = 4.0/TAN(SWPLE) /(1.-CO) * BETA
      TOCR    = TOC
      IF( A1(234).EQ.9 ) TOCR = 0.0
C
  110 CLVOFM  = DLNT(ABETA, CO, XAB, YCO, FKVOFM, 6,8,6,2,2)
      TS2     = TAN(SWPLE)**2
      FM      = SQRT((1. + TS2)/(BETA**2 + TS2))
      CLVA    = CLVOFM * FM * SEXW/SREF
C
      ATAB(1)= 0.0
      DO 120 I = 1, 13
      IF( I.GT.1 ) ATAB(I) = ATAB(I-1) + 3.0
      ANG     = ATAB(I)* 0.01745
      DCLP    = 0.0
      IF( ATAB(I).LE.AS ) GO TO 115
C
C     TIP VORTEX EFFECT
C
      HOB     = (ATAB(I) -AS)/114.6 /AR    *1.5 *(TR+TR*TR)/(1.+TR+TR*TR)
      XEP     = 1.0014 -1.969*HOB +3.0021*HOB**2 -2.0072*HOB**3
      IF( HOB.LE.0.0 ) XEP = 1.0
      DCLEP   = (Z +SQRT(1.+Z*Z))/(XEP*Z + SQRT(1.+ (XEP*Z)**2))
      DCLPA   = (DCLEP -1.) * CLAW * 57.3
      DCLP    = DCLPA * SIN(ANG)
  115 CONTINUE
C     LEADING EDGE EFFECT (RA)
      RA      = DLNT(ANG, TOCR, XANG, YRTOC, FRA, 10, 7, 10, 2,2)
C
C     VORTEX BREAKDOWN EFFECT (FVL)
      X       = AR
      FVL     = 0.0
      IF( X.LT.3.37 ) FVL = DLNT(X,ANG, XX, XY, XF, 12, 7, 12, 4,2)
C
      CLV     = (1.-RA) * FVL * CLVA * SIN(ANG)**2 * COS(ANG)
      IF( CLV.GT.CLVM2 ) CLV = CLVM2
      CLP     = CLPA * SIN(ANG) * COS(ANG)**2
      CLTAB(I) = DCLO + CLV + CLP + DCLP
      IF( KPRINT(17).EQ.2.AND.I.EQ.1 ) WRITE(6,1002)
```

```
      IF( KPRINT(17).EQ.2 ) WRITE(6,1001) CLTAB(I), ATAB(I), RA,
     1              CLP, CLV, DCLP
C
  120 CONTINUE
C
      IF( KPRT.GT.0 ) WRITE(6,1003) CL, ALPHA, AR, ARLO, Z, HOB, DCLPA,
     1                 CLPA, DCLO, CLVM1, CLVM2, TOCR, CLVA, RA, FVL,
     2                 ANG
C
  125 IF( CLVM1.LE.0.0 ) GO TO 20
  130 CALL LNTP(CL, ALPHA, CLTAB, ATAB, 13, 4)
C
      GO TO 200
C
C     CALCULATES ANGLE OF ATTACK AT HIGH LIFT BY MODIFIED
C     WINSTAN METHOD.
C
C
  290 CONTINUE
  300 BETA   = SQRT(1. - SPEED**2)
      BETAN  = BETA * TAN(SWLEI)
      A      = DLNT(BETAN,SWEEP, XIN13,YIN13,ZOUT13, 10,7,10, 2,2)
      EN     = DLNT(BETAN,SWEEP, XIN13,YIN13,ZOUT14, 10,7,10, 2,2)
C
C
      ACLARU = CLDB/CLA + ALO
      EBRK   = YIX/(YIX + YOX)
      CLB    = A * ACLARU**EN * EBRK/ARI * CLAW * 57.29578
  310 IF( CL.LE.CLDB ) GO TO 200
      CLT    = CLB + CL - CLDB
      ALPHA  = (CLT * ARI/(A * CLAW * 57.29578 * EBRK))**(1./EN)
C
      IF( KPRT.GT.0 ) WRITE(6,1004) CL, ALPHA, AR, A, EN, ACLARU,
     1                 EBRK, CLB, CLDB, CLAW
C
  200 CONTINUE
      IF( ALPHA.LT.6.0 ) GO TO 205
      CDLA   = CL * TAN(ALPHA/57.296)
      IF( CDLA.LT.CDL ) CDL = CDLA
  205 RETURN
C
 1000 FORMAT(5X, *AERA DUMP   HIGH ASPECT RATIO* /(1X,7F15.5) )
 1003 FORMAT(5X, *AERA DUMP   LOW ASPECT RATIO* /(1X,7F15.5) )
 1004 FORMAT(5X, *AERA DUMP   CRANKED WING* /(1X,7F15.5) )
 1001 FORMAT(5X,*VORTEX LIFT*, 6F15.5)
 1002 FORMAT(///5X,*VORTEX LIFT*, 8X,*CL*,12X,*ALPHA*,13X,*R*, 13X,
     1          *CLP*, 11X,*CLV*, 11X, *DCLP*, / )
      ENTRY AERA1
  400 KPRT   = KPRINT(17) - 1
      ALPHA  = CL/CLA + ALO
      IF( SPEED.GT.1.0 ) RETURN
      IF( NPNLS.GT.1 ) GO TO 310
      IF( AR.LE.ARLO.AND.CLVM1.GT.0.0 ) GO TO 125
      GO TO 20
C
      END
```

CC = 00168

```
      SUBROUTINE AFTCD(ALPHA, CDAFT)
C
C     CCMPUTE DRAG INCREMENT DUE TO FUSELAGE UPSWEEP
C
      COMMON /BLKA02/ A1(433)
C
      DIMENSION X(6), Y(4), Z(6,4)
C
      DATA  X   / -5., 0., 5., 10., 15., 20. /,
     1      Y   / 0., 5., 10., 15. /,
     2      Z   / 2*0.0, .01, .045, .125, .28, 0.,.0005,.013,.054,.142,
     3            .33, 0.,.004,.0175,.065,.172,.41, 0.,.014,.023,
     4            .085, .215, .53 /
C
      WINC   = A1(280)
      BFUS   = A1(241)
      AB     = A1(243)
      IF( AB.EQ.0.0 ) AB = 1.0
      BAMX   = A1(65)
      SREF   = A1(1)
C
      CDAFT  = 0.0
      IF( BFUS.EQ.0.0 ) RETURN
      AFUS   = ALPHA - WINC
      BMA    = BFUS - AFUS
C
      DCD    = DLNT(BMA, AFUS, X,Y,Z, 6,4,6, 4,2)
      CDAFT  = DCD * BAMX/SREF * AB/1.5
C
      RETURN
      END
```

CC = 00031

```
      SUBROUTINE TORG(ITRIM, SPEED, DCLT, DCDT)
C
C     COMPUTES EFFECT OF TRIM
C
      COMMON /BLKA02/ A3(280), XLE(11), A4(142)
C
      COMMON /BLKA03/ A1(10), XCG, ZCG, CMAC, A2(7)
C
      COMMON /BLKG01/ G1(82), CB, YB, XB, CRX, G2(22), XH, OMEGA,
     1                G3(90)
C
      COMMON /BLKC01/ CL,CD,CM,ALPHA,C1(6),DH,FK,DELCL,CMO,DCMCL,XACWB,
     1                C2(39), CLAT, AH, BH, HSTAR, CH, ABREAK, CLDH,
     2                CLPB, CLDB, CLMAX, C3(22), XACS, C4(12)
C
      DCLT = 0.0
      DCDT = 0.0
C
      CGOCR  = (XCG - XLE(1))/CRX
      XAC    = XACWB
      IF( CL.GT.CLDB ) XAC = XACWB + (XACS - XACWB) * ((CL-CLDB)/
     1                         (CLMAX-CLDB))**2
      IF( CL.GT.CLMAX ) XAC = XACS
      IF( SPEED.GE.1.0 ) XAC = XACWB
      DCMCL  = (CGOCR - XACWB) * CRX/CMAC
      IF( ITRIM.EQ.1 ) DH = 0.0
C
      CLTAIL = CLAT * (ALPHA - HSTAR) + CLDH * DH
      CLWB   = CL - CLTAIL
      XLT    = XH *COS(OMEGA - ALPHA/57.3) /CMAC
C
      CM     = CMO + (CGOCR - XAC) *CRX/CMAC * CLWB - CLTAIL * XLT
      IF( CLDH.EQ.0.0.OR.XLT.EQ.0.0 ) GO TO 60
      IF( ITRIM.EQ.1 ) DH = CM /(CLDH * XLT)
      IF( ITRIM.EQ.1 ) CM = 0.0
C
   60 CONTINUE
C
      IF( DH.EQ.0.0 ) GO TO 100
      DCLT   = DH*CLDH
C
      DCDT   = AH*DH**2 + BH*DH*(ALPHA -HSTAR)
      IF(ALPHA.GT.ABREAK) DCDT = DCDT  + CH*DH*(ALPHA-ABREAK)
C
  100 RETURN
      END
```

CC = 00046

```
      SUBROUTINE DMIN(SPEED, RNOFT, CDMIN)
C
C     MINIMUM DRAG
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKA02/ SREF, A1(103), BASE(10), A2(319)
C
      COMMON /BLKC01/ C1(18), TCD(5), CDFUS(5), CDBOD(5), C2(59),
     1                CDMISC, C3(7)
C
C
      CDFUS(5) = 0.0
      CDBOD(5) = 0.0
C
      FMISC  = A2(128)
C
      CALL FDRG(SPEED, RNOFT)
C
      CDMISC = (TCD(1) + TCD(2) + TCD(3)) * FMISC * 0.01
      CALL ADJUST(2,0, SPEED, CDMISC)
C
      DO 20 I = 1, NBODYS
      CALL BDRG(SPEED, BASE(I), SREF, CDB)
      IF( I.EQ.1 ) CDFUS(5) = CDB
      IF( I.GT.1 ) CDBOD(5) = CDBOD(5) + CDB
   20 CONTINUE
C
      TCD(5) = CDFUS(5) + CDBOD(5)
C
      CALL WDRG(SPEED)
C
      CDMIN  = TCD(1) + TCD(2) + TCD(3) + TCD(4) + TCD(5)
      CDMIN  = COMIN + CDMISC
C
C
   30 RETURN
      END
```

CC = 00038

```
      SUBROUTINE FDRG(SPEED, RNOFT)
C
C     THIS SUBROUTINE CALCULATES FRICTION, FORM, AND INTERFERENCE
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NJET, NVT, ISWP, NPNLS
      COMMON /BLKA02/ SREF, AR, TAPR, SWPLE,
     1                BLEN(10), BWID(10), BHGT(10), BAWET(10), BQ(10),
     2                BNO(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),
     3                BASE(10),  ELEN(10), EWID(10), EHGT(10),
     4                EAWET(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),
     5                ELBT(10), EQF(10), ENO(10), CBAR(10), TW,
     5                XLEW, YWW, YB, CR, BO2, BFUS, FMISC, AB, AFTAW,
     6                CAM(10), TOC(10), AWET(10), SWMT, SPLAN, CONCL,
     7                TWIST, FTWIST, WINC, XLE(11), CRW(11), YW(11),
     8                XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC,
     9                SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),
     1                SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),
     2                SCR(10), HTLF, HTY, HTZ, HTINC
C
      COMMON /BLKC01/ C1(18), TCD(5), CDFUS(5), CDBOD(5), CDNAC(4),
     1                CDWING(4), CDHT(4), CDVT(4), CDSURF(4), C2(47)
C
      COMMON /BLKOV3/ OV3A(7), J, OV3B(4)
C
      COMMON /BLKSUR/ SUR(162), TRB(5,10), TRN(5,10), TRU(5,10),
     1                TRL(5,10), TRS(5,10)
      COMMON /BLKG01/ FRBOD(10), G1(20), FRNAC(10), G2(73),
     1                SWPMT, CBAR2, CLDS, TOCS, SWET, G3(82)
C
      CRITM  = C2(19)
      XTR    = 0.0
C
      DO 10 I=1,3
      TCD(I)    = 0.0
      CDFUS(I) = 0.0
      CDBOD(I) = 0.0
      CDNAC(I) = 0.0
      CDWING(I)= 0.0
      CDHT(I)  = 0.0
      CDVT(I)  = 0.0
      CDSURF(I)= 0.0
   10 CONTINUE
C
C     BODY CONTRIBUTIONS
      IF (NBODYS.EQ.0) GO TO 30
      DO 20 I =1,NBODYS
      IF( J.GT.0 ) XTR = TRB(J,I) * BLEN(I)
      CALL CFEQ(RNOFT,SPEED,BLEN(I),XTR,CDF)
      CALL FFACT(1, FRBOD(I), 0, 0, SPEED, 0.0, FF)
      DRAG = CDF*BAWET(I)/SREF
      IF (I.EQ.1) GO TO 25
      FI = BQ(I)
      IF( SPEED.GT.1.0 ) FI = 1.
      CDBOD(1) = CDBOD(1) + DRAG
      CDBOD(2) = CDBOD(2) + DRAG*(FF -1.)
      CDBOD(3) = CDBOD(3) + DRAG*FF*(FI - 1.)
```

```
      GO TO 20
   25 REFUS = RNOFT*BLEN(1)
      CALL IFACT(1, REFUS, CRITM, SPEED, FI)
      CDFUS(1) =  DRAG
      CDFUS(2) =  DRAG*(FF -1.)
      CDFUS(3) =  DRAG*(FF)*(FI -1.)
   20 CONTINUE
C
C     NACELLE CONTRIBUTIONS
   30 IF (NNACS.EQ.0) GO TO 40
      DO 35 I =1,NNACS
      IF( J.GT.0 ) XTR = TRN(J,I) * ELEN(I)
      CALL CFEQ(RNOFT,SPEED,ELEN(I),XTR,CDF)
      CALL FFACT(2, FRNAC(I), 0, 0, SPEED, 0.0, FF)
      DRAG = CDF*EAWET(I)/SREF
      FI     = EQF(I)
      IF( SPEED.GT.1. ) FI = 1.
      CDNAC(1)      = CDNAC(1) + DRAG
      CDNAC(2)      = CDNAC(2) + DRAG*(FF -1.)
      CDNAC(3)      = CDNAC(3) + DRAG*FF*(FI-1.)
   35 CONTINUE
C
C     WING CONTRIBUTION
   40 IF (NPNLS.EQ.0) GO TO 50
      DO 45 I = 1, NPNLS
      IF( J.GT.0 ) XTR = TRU(J,I) * CBAR(I)
      CALL CFEQ(RNOFT,SPEED, CBAR(I), XTR, CDFU)
      IF( J.GT.0 ) XTR = TRL(J,I) * CBAR(I)
      CALL CFEQ(RNOFT,SPEED, CBAR(I), XTR, CDFL)
      CDF    = 0.5 * (CDFU + CDFL)
C
      CALL FFACT(3, TOC(I), TW, CAM(I), SPEED, CRITM, FF)
      IF( I.EQ.1 ) CALL IFACT(2, SWMT, CRITM, SPEED, FI)
      DRAG = CDF*AWET(I)/SREF
C
      IF( ISWP.GT.0.AND.I.EQ.NPNLS ) CALL CFEQ(RNOFT,SPEED,CBAR2,
     1                 0.0,CDF)
      IF( ISWP.GT.0.AND.I.EQ.NPNLS ) CALL FFACT(3,TOCS,TW,CLDS,
     1                  SPEED,CRITM,FF)
      IF( ISWP.GT.0 ) CALL IFACT(2, SWPMT, CRITM, SPEED, FI)
      IF( ISWP.GT.0.AND.I.EQ.NPNLS ) DRAG = CDF * SWET / SREF
C
      CDWING(1)=  CDWING(1) + DRAG
      CDWING(2)=  CDWING(2) + DRAG*(FF-1.)
      CDWING(3)=  CDWING(3) + DRAG*FF*(FI-1.)
   45 CONTINUE
C
C     SURFACE CONTRIBUTIONS
   50 IF (NSURFS.EQ.1) GO TO 60
      NSURF1 = NSURFS - 1
      DO 55 I = 1, NSURF1
      IF( J.GT.0 ) XTR = TRS(J,I) * SBAR(I)
      CALL CFEQ(RNOFT, SPEED, SBAR(I), XTR, CDF)
      CALL FFACT(3, STOC(I), TS(I), SCAM(I), SPEED, 0.0, FF)
      CALL IFACT(3, SMTSW(I), CRITM, SPEED, FI)
      IF( SPEED.LE.1. ) FI = FI * SHF(I)
```

```
      DRAG = CDF*SAWET(I)/SREF
      IF ( I. EQ. 1.AND. NHT.GT.0) GO TO 100
  200 IF ( NHT.EQ.0 ) GO TO 260
      IF ( I.EQ.2.AND.NVT.GT.0) GO TO 101
  201 IF (NVT.EQ.0) GO TO 270
      IF ( I.GT.2 ) GO TO 102
      GO TO 55
  260 IF ( I.EQ.1.AND.NVT.GT.0) GO TO 101
      IF ( NVT.EQ.0 ) GO TO 102
  270 IF ( I.GT.1 ) GO TO 102
      GO TO 55
  100 CDHT(1) = DRAG
      CDHT(2) = DRAG*(FF-1.)
      CDHT(3) = DRAG*FF*(FI - 1.)
      GO TO 200
  101 CDVT(1) = DRAG
      CDVT(2) = DRAG*(FF-1.)
      CDVT(3) = DRAG*FF*(FI - 1.)
      GO TO 201
  102 CDSURF(1) = CDSURF(1) + DRAG
      CDSURF(2) = CDSURF(2) + DRAG*(FF-1.)
      CDSURF(3) = CDSURF(3) + DRAG*FF*(FI - 1.)
      GO TO 55
   55 CONTINUE
   60 CONTINUE
      DO 70 I = 1,3
      TCD(I) = CDFUS(I) + CDBOD(I) + CDNAC(I) +CDWING(I)
     1        +CDHT(I) + CDVT(I)  + CDSURF(I)
   70 CONTINUE
      RETURN
      END
```

CC = 00143

```
      SUBROUTINE CFEQ(RNOFT,ZMACH,CBAR,XTR,CF)
C
C     THIS SUBROUTINE CALCULATES THE SKIN FRICTION COEFFICIENT
C     USING THE WHITE-CHRISTOPH TECHNIQUE.
C
      COMMON /BLKPR1/ KPRINT(50)
      COMMON /BLKA03/ ROUGHK, A1(19)
C
      FTURB(X) = T*F**2*0.430/(ALOG10(RNL*X*T**1.67*F))**2.56
C
      FLAM(Y)  = 1.328*CFCFIL/SQRT(Y*RNL)
C
      FPRIM(X) = 0.43*T*F*F*((ALOG10(RNL   *X*T**1.67*F))**(-2.56)
     1           -1.11178*(ALOG10(RNL   *X*T**1.67*F))**(-3.56))
C
      IF( KPRINT(19).GT.0 ) WRITE(6,1001) RNOFT,ZMACH,CBAR,XTR,ROUGHK
      IF( CBAR.LE.0.0.OR.CBAR.GT.10000.) GO TO 500
C
      ZMACH2 = ZMACH*ZMACH
      T  =1.0/(1.0+ 0.178*ZMACH2)
      F  =1.0 + 0.03916*ZMACH2*T
      DXNP1  = 0.0
      RNCO   = 0.0
      RNL    = RNOFT
      IF( ROUGHK.LE.0.0 ) GO TO 5
C
      RNL     = RNOFT * CBAR
      AK1= 37.587 + 4.615*ZMACH+2.949*ZMACH2+4.132*ZMACH*ZMACH2
      RNCO    = AK1*(CBAR*12./ROUGHK)**1.0489
      IF (RNL.GT.RNCO) RNL = RNCO
      RNL     = RNL/CBAR
    5 CONTINUE
C
      CFCFIL   = (1.0+ 0.1256*ZMACH2)**(-.12)
      IF (XTR.LE.0.0) GO TO 100
      DXN  = 0.1 * XTR
C
   10 DP       = FPRIM(DXN)
      IF (DP.EQ.0.0) GO TO 200
      DXNP1    = DXN - (DXN * FTURB(DXN) - XTR * FLAM(XTR))/DP
      IF ( DXNP1.LE.0.0) DXNP1 = 0.5 * DXN
      DX       = ABS(DXNP1 - DXN)
      DY       = 0.0001*CBAR
      IF (DX.LE.DY) GO TO 20
      DXN      = DXNP1
      GO TO 10
C
   20 XP       = DXNP1 + CBAR - XTR
      CF       = (XP/CBAR)*FTURB(XP)
      GO TO 300
C
  100 CF       = FTURB(CBAR)
      GO TO 300
C
  200 WRITE (6,1000) DXN, RNL
      CF       = 0.0
```

```
  300 CFBAR = FTURB(CBAR)                                               R1T3965
      IF ( KPRINT(19).EQ.0) GO TO 400                                   R1T3966
      WRITE(6,1003) CF, CFBAR, DXNP1, CFCFIL, RNL, RNCO                 R1T3967
C                                                                       R1T3968
  400 RETURN                                                            R1T3969
  500 CF      = 0.0                                                     R1T3970
      WRITE(6,1002) CBAR                                                R1T3971
      RETURN                                                            R1T3972
C                                                                       R1T3973
 1000 FORMAT (10X*SUBROUTINE CFEQ WILL NOT CONVERGE*/ 1X,3F15.7 )       R1T3974
 1001 FORMAT(10X,*CFEQ INPUT*, 1PE15.4,0P4F15.7  )                      R1T3975
 1002 FORMAT(10X,*CBAR =*,1PE14.5,2X,*OUT OF RANGE IN CFEQ, CF SET EQUALR1T3976
     1 TO ZERO* // )                                                    R1T3977
 1003 FORMAT(10X,*CFEQ OUTPUT*, 4F15.7, 1P2E15.4 )                      R1T3978
      END                                                               R1T3979

                                                          CC = 00071
```

```
      SUBROUTINE FFACT(ID,GEOM, TYP,CLD,SPEED,CRITM,FF)
C
C     THIS SUBROUTINE CALCULATES FORM FACTORS FOR EACH COMPONENT.
C
      ITYP   = TYP
      FF = 1.0
      IF (SPEED.GT.1.0) GO TO 40
C
      GO TO (10,20,30) ID
C
C     FUSELAGE OR BODIES (ID=1)
   10 FR = GEOM
      FF = 1.0 + 60./FR**3.0 + 0.0025*FR
      GO TO 40
C
C     NACELLE (ID=2)
   20 FR = GEOM
      FF = 1.0 + 0.35/FR
      GO TO 40
C
C     WING AND SURFACES (ID=3)
   30 TOC = GEOM
      TOC2= TOC*TOC
      IF (ITYP.LE. 7) FF = 1.0 + 1.44*TOC + 2.0*TOC2
      IF (ITYP.EQ. 9) FF = 1.0 + 1.2 *TOC + 100.*TOC2*TOC2
      IF (ITYP.GE. 10.AND. ITYP.LE.20) FF = 1.0 + 1.68*TOC +3.*TOC2
      IF (ITYP.NE. 8) GO TO 40
      DELTAM = SPEED - CRITM
      IF (DELTAM.GE.0.0) ZK = 0.202
      IF (DELTAM.LE.-.2) ZK = 0.12
      IF (DELTAM.GT.-.2.AND.DELTAM.LT.0.0) ZK = 0.202 + .8972*DELTAM
     1     +2.1944*DELTAM**2 -1.2340*DELTAM**3
      FF =ZK*CLD/.4+1.44*TOC + 2.0*TOC2 +1.
C
   40 RETURN
      END
```

CC = 00036

```
      SUBROUTINE IFACT(ID,PARAM,CRITM,SPEED,FI)
C
C     THIS SUBROUTINE CALCULATES INTERFERENCE FACTORS FOR THE
C     FUSELAGE AND LIFTING SURFACES.
C
      DIMENSION XMACH(7),REFUS(9), WBR(63)
C
      DATA    XMACH /      0.25,0.4,0.6,0.7,0.8,0.85,0.9 /,
     1        REFUS / 3.0,10.,20.,30.,40.,60.,80.,200.,100000./,
     1        WBR   / 1.063,1.02,.979,.955,.926,.902,.867,
     1                1.076,1.036,.996,.972,.942,.92,.884,
     1                1.066,1.059,1.02,.996,.966,.943,.91,
     1                1.044,1.049,1.037,1.013,.988,.968,.939,
     1                0.993,1.018,1.032,1.015,1.003,.993,.973,
     1                0.940,0.992,1.018,1.014,1.014,1.01,1.007,
     1                0.932,0.984,5*1.015,
     1                0.924,0.976,5*1.015,0.924,0.976,5*1.015 /
C
      FI = 1.0
      IF (SPEED.GT.1.0) GO TO 30
C
      GO TO (10,20,20) ID
C
C     FUSELAGE CORRELATION FACTOR (RWB)
   10 PAR10 = PARAM/10.0**6
      XM1    = CRITM - .1
      XM     = SPEED
      IF( XM.GT.XM1 ) XM = XM1
C
      FI     = DLNT( XM, PAR10, XMACH, REFUS, WBR, 7,9,7,2,2)
      RWB = FI
      GO TO 30
C
C     LIFTING SURFACES
   20 A = COS(PARAM)
      IF (A.GE.0.75) RLS = -.9946 + 5.3568*A -4.5389*A*A +1.2487*A**3
      IF (A.GT.0.65.AND.A.LT.0.75) RLS=.9966 -.752*(.75-A)
      IF (A.LE.0.65) RLS = 0.9214
      B = SPEED - 0.25
      IF (B.LT.0.0) B=0.0
      RLS = RLS - .0015 + .1818*B - .2756*B*B + 1.0677*B**3
      FI = RLS*RWB
      IF (ID.EQ.3) FI = RLS
C
   30 RETURN
      END
```

CC = 00046

```
      SUBROUTINE WDRG(FMACH)
C
C     COMPUTES TOTAL AIRPLANE WAVE DRAG
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKA02/ SREF, A1(53), BNO(10),
     1                BAMX(10), BABS(10), BLNS(10),
     2                BLBT(10), BASE(10), E1(40), EAMX(10), EIN(10),
     3                EXIT(10), ELNS(10), ELBT(10), EQF(10), ENO(10),
     4                W1(95), S1(30), STOC(10), S2(30),
     5                SWL(10), SWT(10), STAPR(10), SCR(10), S3(4)
C
      COMMON /BLKG01/ FRBOD(10),ARS(10),SSEX(10),FRNAC(10), SWPLE,
     1                SWPOC, SWPMC, SWPTE, DOB, TOCW, CLD, SEXW, B1(31),
     2                AR, TAPR, SPLAN, G1(20), ESWLE, ESWTE, G2(96)
C
      COMMON /BLKWPD/ WPD(9)
C
      COMMON /BLKBTA/ BETA
C
      COMMON /BLKC01/ C1(18), TCD(5), CDFUS(5), CDBOD(5), CDNAC(4),
     1                CDWING(4), CDHT(4), CDVT(4), CDSURF(4), C2(47)
C
C
      TCD(4) = 0.0
      CDFUS(4) = 0.0
      CDBOD(4) = 0.0
      CDNAC(4) = 0.0
      CDWING(4)= 0.0
      CDHT(4)  = 0.0
      CDVT(4)  = 0.0
      CDSURF(4)= 0.0
C
      IF( FMACH.LE.1.0 ) RETURN
      BETA   = SQRT( FMACH**2 -1.)
C
C     BODY CONTRIBUTIONS
      IF( NBODYS.EQ.0 ) GO TO 100
      DO 50 I = 1, NBODYS
C
      CALL COWN(BAMX(I), BLNS(I), 0.0, CDW1)
      REX    = SQRT(BABS(I)/3.14159)
      CALL COWT(BAMX(I), BLBT(I), REX, CDW2)
C
      IF( I.EQ.1 ) COFUS(4) = (CDW1 + CDW2) * BAMX(1)/SREF
      IF( BNO(I).EQ.0.0 ) BNO(I) = 1.0
      IF( I.GT.1 ) CDBOD(4) = CDBOD(4) + (CDW1+CDW2)*BAMX(I)/SREF*BNO(I)
   50 CONTINUE
C
C     NACELLE CONTRIBUTION
  100 IF( NNACS.EQ.0 ) GO TO 200
C
      DO 150 I =1, NNACS
      RIN    = SQRT(EIN(I)/3.141593)
      REX    = SQRT(EXIT(I)/3.141593)
```

```
C
      CALL CDWN(EAMX(I), ELNS(I), RIN, CDW1)
      CALL CDWN(EAMX(I), ELBT(I), REX, CDW2)
C
      IF( ENO(I).EQ.0.0 ) ENO(I) = 1.0
      CDNAC(4) = (CDW1 + CDW2) * EAMX(I)/SREF * ENO(I) + CDNAC(4)
  150 CONTINUE
C
C     WING CONTRIBUTION
  200 WPD(1) = AR * (1.-DOB)**2 * SPLAN/SEXW
      WPD(2) = TAPR/(1.-DOB*(1.-TAPR))
      WPD(3) = ESWLE
      WPD(4) = ESWTE
      WPD(5) = FMACH
      WPD(6) = SEXW/SREF
      WPD(7) = W1(11)
      WPD(8) = CLD
      WPD(9) = TOCW
C
      CALL CDWW(CDW1)
      CDWING(4) = CDW1
C
  250 IF( NSURFS.LE.1 ) GO TO 300
      DO 290 J = 2, NSURFS
      I        = J-1
C
      WPD(1) = ARS(I)
      WPD(2) = STAPR(I)
      WPD(3) = SWL(I)
      WPD(4) = SWT(I)
      WPD(6) = SSEX(I)/SREF
      WPD(7) = S1(10+I)
      WPD(8) = S1(20+I)
      WPD(9) = STOC(I)
C
      CALL CDWW(CDW1)
      IF( I.EQ.1.AND.NHT.GT.0 ) CDHT(4) = CDW1
      IF( NHT.EQ.0 ) GO TO 260
      IF( I.EQ.2.AND.NVT.GT.0 ) CDVT(4) = CDW1
      IF( NVT.EQ.0 ) GO TO 270
      IF( I.GT.2 ) CDSURF(4) = CDSURF(4) + CDW1
      GO TO 290
C
  260 IF( I.EQ.1.AND.NVT.GT.0 ) CDVT(4) = CDW1
      IF( NVT.EQ.0 ) GO TO 280
  270 IF( I.GT.1 ) CDSURF(4) = CDSURF(4) + CDW1
      GO TO 290
C
  280 CDSURF(4) = CDSURF(4) + CDW1
  290 CONTINUE
C
  300 TCD(4) = CDFUS(4) +CDBOD(4) +CDNAC(4) +CDWING(4) +CDHT(4)
     1            +CDVT(4) + CDSURF(4)

C
      RETURN
```

END

R1T4177

CC = 00113

```
      SUBROUTINE COWN(AMAX, XLNOS, RIN, CDW)                            R1T4179
C                                                                       R1T4180
C     NOSE WAVE DRAG (BODY COMPONENT)                                   R1T4181
C                                                                       R1T4182
      COMMON /BLKBTA/ BETA                                              R1T4183
C                                                                       R1T4184
      DIMENSION XLD(7), YBETA(4), FCDW(7,4)                             R1T4185
C                                                                       R1T4186
      DATA  XLD / 1., 2., 3., 4., 6., 8., 12. /,                        R1T4187
     1      YBETA/ 0.0, 0.32,  .458, 0.6633 /,                          R1T4188
     2      FCDW / .1656,.0617,.0373,.0287,.0198,.0149,.0083,           R1T4189
     3             .2376, .1256, .088, .061, .0308, .0177, .0082,       R1T4190
     4             .2952, .1667, .1067, .066, .0311, .0175,.0081,       R1T4191
     5             .3816, .2010, .1031, .0624,.0297, .0172,.0079 /      R1T4192
C                                                                       R1T4193
      CDW   = 0.0                                                       R1T4194
      IF( AMAX.LE.0.0) GO TO 50                                         R1T4195
      DMAX=SQRT(AMAX/0.785398 )                                         R1T4196
      ELOD=XLNOS/DMAX                                                   R1T4197
      IF(RIN.GT.0.0) GO TO 40                                           R1T4198
      IF( ELOD.LE.0.0 ) GO TO 50                                        R1T4199
C                                                                       R1T4200
      IF( BETA.GE.0.6633 ) GO TO 20                                     R1T4201
C                                                                       R1T4202
   10 CDW    = DLNT(ELOD,BETA, XLD, YBETA, FCDW, 7,4,7, 2,2)            R1T4203
      GO TO 50                                                          R1T4204
C                                                                       R1T4205
   20 X      = BETA/SQRT(1.+ ELOD**2)                                   R1T4206
      CDW    = (1.2 + 1.15 * X)/(1. + 1.9 * X) /(1. + ELOD**2)          R1T4207
      GO TO 50                                                          R1T4208
C                                                                       R1T4209
C                                                                       R1T4210
C     FOR A BODY WITH AN OPEN NOSE, SUCH AS A NACELLE, THE CONTRIBUTION R1T4211
C     OF THE NOSE TO THE TOTAL BODY WAVE DRAG COEFFICIENT IS COMPUTED ASR1T4212
C     INDICATED BELOW. (THE EXPRESSIONS USED FOR OPEN AND CLOSED BODY   R1T4213
C     BOATTAIL CONTRIBUTIONS DO NOT REQUIRE THIS DISTINCTION--SEE CDWT.)R1T4214
C                                                                       R1T4215
   40 SQRTB  = SQRT(BETA)                                               R1T4216
      IF( SQRTB.LT.0.8144 ) SQRTB = 0.8144                              R1T4217
      CDW    = ((1.-2.*RIN/DMAX)/ELOD)**1.5/SQRTB                       R1T4218
C                                                                       R1T4219
   50 RETURN                                                            R1T4220
      END                                                               R1T4221

                                                          CC = 00043
```

```
      SUBROUTINE CDWT(AMAX, XLAFT, REX, CDW)
C
C     BOATTAIL WAVE DRAG (BODY COMPONENT)
C
      COMMON /BLKBTA/ BETA
C
      DIMENSION DBOD(5), CDWBT(5)
C
      CDW   = 0.0
      IF( AMAX.LE.0.0) RETURN
      IF (XLAFT .LE. 0.0001) RETURN
      DMAX=SQRT(AMAX/0.785398 )
      ELOD=XLAFT/DMAX
      X=BETA/ELOD
      Y=X*X
      Z=ELOD*ELOD
C
C
C     BOATTAIL WAVE DRAG IS A FUNCTION OF BOATTAIL FINENESS RATIO, BASE
C     DIAMETER/MAXIMUM DIAMETER, AND MACH NUMBER.  COMPUTE THE BOATTAIL
C     WAVE DRAG COEFFICIENT AT FOUR  DBOD(I)  VALUES AND INTERPOLATE TO
C
      DBOD(1)=0.0
      CDWBT(1) = (1.165 -.5112*X -.5372*Y +.3964*X*Y)/Z
      IF( X.GT.1.0 ) CDWBT(1) = 0.513/X/Z
C
      DBOD(2)=0.4
      CDWBT(2) = (1.067 -1.709*X +1.6632*Y -.686*X*Y)/Z
      IF( X.GT.1.0 ) CDWBT(2) = 0.3352/X/Z
C
      DBOD(3)=0.6
      CDWBT(3) = (0.7346 -1.4618*X +1.5795*Y -.6542*X*Y)/Z
      IF( X.GT.1.0 ) CDWBT(3) = 0.1980/X/Z
C
      DBOD(4)=0.8
      CDWBT(4) = (0.2555 -.5008*X +.5024*Y -.2077*X*Y)/Z
      IF( X.GT.1.0 ) CDWBT(4) = 0.0494/X/Z
C
      CDWBT(5)=0.0
      DBOD(5)=1.0
      BDOD=2.0*REX/DMAX
      CALL LNTP(BDOD, CDW, DBOD, CDWBT, 5, 2)
C
      RETURN
      END
```

CC = 00045

```
      SUBROUTINE CDWW(CDOSR)                                            R1T4269
C     PROCEDURE FOR WING PRESSURE DRAG                                  R1T4270
C                                                                       R1T4271
      COMMON /BLKWPD/ AR, ZLAM, ZLE, ZTE, ZM, SOSR, TYPE, CAM, TOC       R1T4272
C                                                                       R1T4273
      COMMON /BLKPRT/ KPRINT(50)                                        R1T4274
      COMMON /BLKBD2/ AA(22), BB(22), CC(22), DD(22), XT(22)            R1T4275
C                                                                       R1T4276
   10 N      = TYPE                                                     R1T4277
C                                                                       R1T4278
      HOT    = CAM * 0.055 /TOC                                         R1T4279
      XTOC=XT(N)                                                        R1T4280
      TOC2= TOC**2                                                      R1T4281
      ROOT=(AA(N)+BB(N)*TOC+CC(N)*TOC2+DD(N)*TOC2*TOC)/TOC               R1T4282
      IF (N.EQ.8) ROOT= 0.88216*(TOC)**1.606/TOC                         R1T4283
C                                                                       R1T4284
C                                                                       R1T4285
C     CALCULATION OF QUANTITIES THAT ARE FUNCTIONS OF GEOMETRY.         R1T4286
C                                                                       R1T4287
   17 ZKW =1.0                                                          R1T4288
      TOC2  = TOC**2                                                    R1T4289
      TOC13 = TOC**0.33333333                                           R1T4290
      TOC53 = TOC**1.6666667                                            R1T4291
      ARW   = AR* TOC13                                                 R1T4292
      ARW3  = TOC*AR**3                                                 R1T4293
      DEN1  = 1.0/ARW+ARW3                                              R1T4294
      DEN2  = 2.0/ARW3+1.0                                              R1T4295
      ZKT    = 1.+4.*(.5-XTOC*(1.+.5*SQRT(ROOT)))**2                    R1T4296
     1          -0.25 * SQRT(ROOT) * (1.-XTOC)**2                       R1T4297
      ZKC = 1.0 +          2.5* HOT **2                                 R1T4298
      ZKB    = 1.069                                                    R1T42981
      IF( ZKW.EQ.1.2 ) ZKB = 1.0                                        R1T42982
      TR1    = 0.5/(1.+ZLAM)**2                                         R1T4299
      TR2    = 1./(1.+ZLAM**2)**2                                       R1T4300
      ZLAM1  = 1. + TR2 * (TAN(ZLE) + TAN(ZTE))**2                      R1T4301
      IF( ZLE.GE.ZTE ) ZKP = (COS(ZLE) + TR1 * (TAN(ZLE)**2             R1T4302
     1                       - TAN(ZTE)**2))/ZLAM1                      R1T4303
      IF( ZTE.GT.ZLE ) ZKP = (COS(ZTE) + TR1 * (TAN(ZTE)**2             R1T4304
     1                       - TAN(ZLE)**2))/ZLAM1                      R1T4305
C                                                                       R1T4306
      ZK    = (2.0/DEN1 + 3.33/DEN2)*ZKP                                R1T4307
      X     = ARW*ZKP                                                   R1T4308
      IF(ARW.GE.1.0) X=SQRT(ZKP/(1.7321*(1.0-ZKP)+1.0/(ZKP*ARW**2)))    R1T4309
C                                                                       R1T4310
C     THIS DO LOOP SOLVES FOR *ARE* BY NEWTON*S METHOD.                 R1T4311
C                                                                       R1T4312
      DO 100 I=1,10                                                     R1T4313
      X3    = X**3                                                      R1T4314
      X4    = X*X3                                                      R1T4315
      F     = (2.0*X)/(X4+1.0) + (3.33*X3)/(X3+2.0) - ZK                R1T4316
      FP    = (2.0-6.0*X4)/((X4+1.0)**2) + (20.0*X**2)/((X3+2.0)**2)    R1T4317
      FFP   = F/FP                                                      R1T4318
      IF(KPRINT(20).EQ.2 .AND. I.EQ.1) WRITE(6,90)                      R1T4319
C                                                                       R1T4320
      IF(KPRINT(20).EQ.2) WRITE(6,95) I, X, F, FP, FFP                  R1T4321
C                                                                       R1T4322
```

```
      X     = X-FFP
      IF (ABS(FFP).LE.0.0001) GO TO 110
  100 CONTINUE
  110 IF(KPRINT(20).EQ.2) WRITE(6,36)
C
      IF(KPRINT(20).EQ.2) WRITE(6,45) ZLE, ZTE, TOC2, TOC13, TOC53,
     1              ARW, ARW3, DEN1, DEN2, ZKT, ZKC, ZLAM1, ZKP
C
C
      BETA2 = ZM**2-1.0
      BETA  = SQRT(BETA2)
      DIV   = BETA/TOC2
      BETAW = BETA/TOC13
      BETAW2= BETAW**2
      BETAW3= BETAW*BETAW2
      CD1   = 0.0
      CD2   = 0.0
      CDA   = TOC2*( 2.0+3.333/(ZKW**2.8))
      ZKLIN = CDA/TOC2
      IF (ZM .GT. 1.0)  CD1 = CDA/BETA
      TEST1 = 1.0/COS(ZLE) + 0.01
      IF (ZM .GT. TEST1)CD2 = CDA/SQRT(BETA2-TAN(ZLE)**2)
      ARE=X
      TEST   = ABS(TAN(ZLE))
      UEXP   = (1.+2.* ZLAM**0.33333)
      FB   = 0.3 + 0.7*ZKP**UEXP
      XM     = 0.5* (1.+ZLAM**2 *(2.-ZLAM)**3 )
      Z      = COS(ZLE) + COS(ZTE)
      IF( BETA.GT.TEST ) XM = (TEST/BETA)**Z
  130 ZML  = SQRT(TEST**2+1.0)
      ARE3 = ARE**3
      ZKW1   = ZKW**3.8
      T1     = 1./ARE /(1.+(1.+ZLAM) * FB * BETAW**(1.+ZKW) )
      T2     = ARE3 /(1.+0.33333 *ARE3 *BETAW**4)
      T3     = 2./ARE3 /(1.+(.66667+ZLAM) *BETAW**(1.+ZKW1) *FB)
      T4     = 1./(1.+3.* ARE * BETAW**4)
      ZFEXP  = FB**XM
      FBXM=FB**XM
      D1     = ZKB * ZKW * BETAW * FBXM +(T1+T2)
      D2     = ZKB * ZKW1 * BETAW * FBXM +(T3+T4)
      CDW    = ZKT * ZKW * ZKC * ZKB * TOC53 * (2./D1 +3.33/D2)
      CDWA   = CDW/TOC53
      CDOSR=CDW*SOSR
      IF (ZM.GT.1.0) CDD= CDW*DIV
      AREW = ARE
      ARE  = AREW/TOC13
C
      IF(KPRINT(20).EQ.2) WRITE(6,47)
C
      IF (KPRINT(20).EQ.2)WRITE(6,48)DIV,BETA2,BETAW,BETAW2,CDA,CD1,CD2,
     1       TEST,XM,FB,ZML,T1,T2,T3,T4,D1,D2,CDW,CDD
C
      VLE    = ZLE * 57.2956
      VTE    = ZTE * 57.2956
      IF (KPRINT(20).GT.0) WRITE(6,25) AR,ZKP,ZLAM,ZKT,VLE,ZKC,VTE,
     1              ZKLIN,TOC,ARW,TYPE,CAM,TOC,TOC13,XTOC,TOC53,ROOT,
```

```
     2               FB, HOT, ZML, ZKW, ARE, SOSR, AREW
C
  550 IF (KPRINT(20).GT.0) WRITE(6,600)
C
      IF (KPRINT(20).EQ.1) WRITE(6,610) ZM,BETAW,XM,ZFEXP,CDWA,CDW,
     1                                  CD1,CD2,CDDSR
C
C
   90 FORMAT(4H0  I,5X,1HX,14X,1HF,14X,2HFP,13X,3HFFP/)
   95 FORMAT(1H ,I3,3X,1P4E15.7)
   36 FORMAT(1H0,2X,3HZLE,12X,3HZTE,12X,4HTOC2,11X,5HTOC13,10X,5HTOC53/
     1        1H ,2X,3HARW,12X,4HARW3,11X,4HDEN1,11X,4HDEN2,11X,3HZKT/
     1            3X,3HZKC,12X,5HZLAM1,10X,3HZKP )
   45 FORMAT(1H0,1P5E15.7/(1X,1P5E15.7))
   47 FORMAT( 1H0,2X,3HDIV,12X,5HBETA2,10X,5HBETAW,10X,6HBETAW2,9X,3HCDA/
     1        1H ,2X,3HCD1,12X,3HCD2,12X,4HTEST,11X,2HXM,13X,2HFB/
     1        1H ,2X,3HZML,12X,2HT1,13X,2HT2,13X,2HT3,13X,2HT4/
     1        1H ,2X,2HD1,13X,2HD2,13X,3HCDW,12X,6HKLINEQ)
   48 FORMAT(1H0,1P5E15.7/(1X,1P5E15.7))
   25 FORMAT(///,16X,36H W I N G    P R E S S U R E   D R A G//
     111X,17H INPUT PARAMETERS,16X,18H OUTPUT PARAMETERS//
     17X,19H ASPECT RATIO    =    ,F7.4,10X,12H KPLANF   =   ,F7.4/
     17X,19H TAPER RATIO     =    ,F7.4,10X,12H KTHICK   =   ,F7.4/
     17X,19H LE SWEEP DEG    =    ,F7.4,10X,12H KCAMB    =   ,F7.4/
     17X,18H TE SWEEP DEG    =    ,F8.4,10X,12H KLINTH   =   ,F7.4/
     17X,19H THICK RATIO     =    ,F7.4,10X,12H ARWIG    =   ,F7.4/
     17X,19H AIRFOIL TYPE    =    ,F3.0,F4.3,F5.3,5X,12H T/C 1/3 =   ,F7.4/
     17X,19H XT/C            =    ,F7.4,10X,12H T/C 5/3 =   ,F7.4/
     17X,19H RO/T            =    ,F7.4,10X,12H F        =   ,F7.4/
     17X,19H H/T             =    ,F7.4,10X,12H M*       =   ,F7.4/
     17X,19H KW              =    ,F7.4,10X,12H ARE      =   ,F7.4/
     17X,19H SW/SREF         =    ,F7.4,10X,12H AREW     =   ,F7.4/)
  600 FORMAT (//31X,8H RESULTS //,72H MACH    BWIG     EXP    FEXP     CDWIG
     1     CD      CDTH2D    CDTH3D    CD/SREF )
  610 FORMAT(1H , 4(F5.3,2X),5(F7.5,2X))
      RETURN
      END
```

CC = 00149

```
      SUBROUTINE CDRG(SPEED, AEROK, DELCL,CDC)
C
C     CAMBER DRAG
C
      COMMON /BLKG01/ G1(47), SEXW, G2(31), AR, TR, SPLAN, G3(118)
C
      COMMON /BLKA02/ SREF, A1(432)
      CDC    = 0.0
      ARSREF = AR * SPLAN/SREF
      E      = 1.0/(3.14159*ARSREF*AEROK)
C
C
      COEF1  = 1.0/(3.14159*ARSREF*(1.0-E))
      COEF2  = 0.7 * SEXW/SREF
      IF( E.GE.1.0 ) GO TO 15
      IF( SPEED.GE.1.0 ) GO TO 5
      IF( COEF1.GT.COEF2 ) GO TO 15
    5 CDC    = COEF1 * DELCL**2
      GO TO 10
   15 CONTINUE
      CDC    = COEF2 * DELCL**2
C
   10 RETURN
      END
```

CC = 00024

```
      SUBROUTINE BDRG(SPEED, AB, SREF, CDB)                             R1T4442
C                                                                       R1T4443
C     BASE DRAG                                                         R1T4444
C                                                                       R1T4445
      COMMON /BLKPRT/ KPRINT(50)                                        R1T4446
C                                                                       R1T4447
      CDB    = 0.0                                                      R1T4448
      IF( AB.LE.0.0 ) RETURN                                            R1T4449
      IF(SPEED.GE.1.0) GO TO 10                                         R1T4450
      CDB    = (0.1 + 0.1222 * SPEED**8) * AB/SREF                      R1T4451
      GO TO 20                                                          R1T4452
C                                                                       R1T4453
   10 CONTINUE                                                          R1T4454
      CDB    = (1.42 /(3.15 + SPEED*SPEED) ) * AB /SREF                 R1T4455
      IF( SPEED.LE.1.8 )  CDB = 0.2222 * AB/SREF                        R1T4456
C                                                                       R1T4457
   20 CONTINUE                                                          R1T4458
      IF(KPRINT(15).EQ.0) GO TO 30                                      R1T4459
      WRITE(6,1000) SPEED, AB, SREF, CDB                                R1T4460
 1000 FORMAT(10X*MACH =*F10.7,5X*BASE AREA =*F10.7,5X*SREF =*           R1T4461
     1                   F12.5,5X*BASE DRAG =*F10.7 / )                 R1T4462
C                                                                       R1T4463
   30 RETURN                                                            R1T4464
      END                                                               R1T4465
                                                            CC = 00024
```

```
      SUBROUTINE CMOW(SPEED, CMO)
C
C     COMPUTES ZERO LIFT PITCHING MOMENT OF WING
C
      COMMON /BLKBD4/ XBD4(5), YBD4(5), ZBD4(3), FBD4(5,5,3)
C
      COMMON /BLKA01/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS
C
      COMMON /BLKA02/ A1(433)
C
      COMMON /BLKC01/ C1(100)
C
      COMMON /BLKG01/ G1(49), SXX(10), G2(42), ESWQC, G3(98)
C
      COMMON /BLKPRT/ KPRINT(50)
C
      DIMENSION CMC4(20)
      DATA CMC4 / 4*-0.25, 3*-0.219, -0.3, 0.0, 11*-0.2066 /
C
C
      FMCR    = C1(72)
      CMOS    = 0.0
      NI      = A1(235)
C
      DO 100 I = 1, NPNLS
      CMOS    = CMOS + A1(244 + I) * SXX(I) * CMC4(NI)
  100 CONTINUE
C
      CMO     = 0.0
      IF( SPEED.GE.1. ) RETURN
C
      ARXR    = G2(35)
      TOC     = G1(46)
      TWIST   = A1(278)
      SEXW    = G1(48)
      CMOS    = CMOS/SEXW
C
      CMOB    = ARXR * COS(ESWQC)**2/(ARXR + 2.*COS(ESWQC)) * CMOS
C
      TRX     = G2(29)/G2(27)
      CALL TLNT(ESWQC,ARXR,TRX, CMOOT, XBD4,YBD4,ZBD4,FBD4, 5,5,3, 5,5)
      CMOT    = CMOOT * TWIST
C
      FMACH   = SPEED
      IF( FMACH.GT.FMCR ) FMACH = FMCR
      CMACH   = (1. + 5.9*TOC*FMACH**5)/SQRT(1.-(FMACH*COS(ESWQC))**2)
C
      CMO     = (CMOB + CMOT) * CMACH
C
      IF( KPRINT(25).GT.0 ) WRITE(6,1000) CMO, CMOB, CMOT, CMACH,
     1                CMOS, ARXR, TOC, TWIST, ESWQC, TRX, CMOOT, FMCR,
     2                ( A1(234+I), A1(244+I), SXX(I), I = 1,NPNLS )
C
 1000 FORMAT(5X, *CMOW DUMP*/(1X,7F15.5) )
C
      RETURN
```

END

RIT4523

CC = 00057

```
      SUBROUTINE WBAC(SPEED, XACR)
C
C     COMPUTES PITCHING MOMENT SLOPE OF WING
C
C
      COMMON /BLKA01/ NBODYS,NNACS,NSURFS,NHT,NVT,ISWP,NPNLS
C
      COMMON /BLKA02/ A02(433)
C
      COMMON /BLKA03/ ROUGHK, CLE(3), CCR(3), YC(3), A03(10)
C
      COMMON /BLKG01/ G1(44), DOB, TOC, CLD, G2(32), AR, TR, SPLAN,
     1                G3(3), CRX, CBX, CTX, YIX, YOX, SIX, SOX, ARI,
     2                ARXR, CBXP, AROP, SOXP, SWPLEI, SWPLEO,
     3                SWPMCI, SWPMCO, G4(99)
C
      COMMON /BLKC01/ C1(100)
C
      COMMON /BLKPRT/ KPRINT(50)
C
      DIMENSION XDB(8), YDOB(8), XBDOL(5), YAON(6), FCP(5,6)
C
      DATA XDB  / 0., .05, .1, .15, .2, .3, .4, .5 /,
     1     YDOB / 0., .1, .154, .19, .219, .266, .3, .33 /
C
      DATA XBDOL / 0.4, .7, 1.0, 1.25, 1.67 /,
     1     YAON  / 0., .4, .8, 1.2, 1.6, 2.0 /,
     2     FCP   / .54,.535,.525,.516,.5, .42,.435,.45,.46,.46,
     3            .35,.377,.4,.414,.425, .295,.33,.355,.375,.394,
     4            .246,.285,.32,.345,.365, .21,.25,.288,.315,.34 /
C
      AR1    = ARXR
      TR1    = CTX/CRX
      TW     = A02(235)
      FMCRO  = C1(72)
C
      IF(NPNLS.GT.1) AR1 = ARI
      IF(NPNLS.GT.1) TR1 = CBX/CRX
C
   20 CALL ACCR(SPEED,AR1,SWPLEI,SWPMCI,TR1,SIX,TOC,TW,FMCRO,XACR,CLAI,
     1                  XACSW)
      XACRX  = XACR
      IF( NPNLS.EQ.1 ) GO TO 100
C
C     CRANKED WING PLANFORMS
C
      TR2    = CTX/CBXP
C
      CALL ACCR(SPEED,AROP,SWPLEO,SWPMCO,TR2,SOXP,TOC,TW,FMCRO,
     1                  XACRP, CLAOP, XACSOP)
C
      XACRO  = XACRP * CBXP/CRX - YIX*0.5*TAN(SWPLEO)/CRX
     1                + YIX * TAN(SWPLEI)/CRX
      XACSO  = XACSOP * CBXP/CRX +XACRO - XACRP*CBXP/CRX
C
      XACR   = (CLAI * SIX * XACRX + CLAOP * SOXP * XACRO )/
```

```
     1             (CLAI * SIX + CLAOP * SOXP)
C
      XACSW  = (CLAI * SIX * XACRX + CLAOP * SOXP * XACSO )/
     1           (CLAI * SIX + CLAOP * SOXP)
  100 XACWB  = XACR
C
C
C     WING-BODY COMBINATION
C
      BLN    = A02(85)
      XLE    = A02(281)
      TANSQC = 0.5 * (TAN(SWPLEI) + TAN(SWPMCI) )
      BO2    = 0.5 * SQRT(AR * SPLAN)
      YWX    = A02(303)
      DIA    = DOB * BO2 * 2.
      IF( SPEED.GE.1.2 ) GO TO 200
C     SUBSONIC CALCULATION OF XACN (NOSE) AND XACBW (WING CARRY-OVER)
      XLEQ   = BLN + 1.6 *(XLE - BLN)
      XACN   = -0.54 * XLEQ/CRX
C
      BARE   = ARXR * SQRT(1.-FMCRO**2)
      IF( SPEED.LT.FMCRO ) BARE = ARXR * SQRT(1.-SPEED**2)
      CALL LNTP(DOB, FDOB, XDB,YDOB, 8,4)
      DXQC   = G4(18)
      XACBW  = 0.25 + DXQC * FDOB/CRX
C
      IF( BARE.GE.4. ) GO TO 190
      XACBWO = 0.125 * ARXR * TAN(SWPLEI) * (1.+CTX/CRX)
      IF( XACBWO.GT.0.5 ) XACBWO = 0.5
C
      XACBW  = (XACBWO - XACBW) * (BARE -4.)**2/16. + XACBW
  190 XACBW1 = XACBW
      XACN1  = XACN
C
      IF( SPEED.LE.FMCRO ) GO TO 290
C
C     SUPERSONIC CALCULATION OF XACN & XACBW
  200 BETA   = 0.663325
      IF( SPEED.GT.1.2 ) BETA = SQRT(SPEED**2 -1.)
C
      AON    = (XLE - BLN)/BLN
      IF( AON.LT.0. ) AON = 0.
      BDOL   = BETA *DIA /BLN
C
C     FIGURE 4.2.2.1-23A DATCOM                    *******
      XCPOL  = DLNT(BDOL, AON, XBDC ,YAON,FCP, 5,6,5, 2,2)
      XACN   = XLE/CRX * (XCPOL -1.)
C
      BDOC   = BETA * DIA/CRX
      BCOT   = BETA * TAN(SWPLEI)
      A1     = 0.5845
      IF( BCOT.GE.1. ) A1 = 0.5985 + 0.00214*ALOG(BCOT)
      IF( BCOT.LT.1.0.AND.BCOT.GT.0.1) A1 = 0.5985 + 0.00607*ALOG(BCOT)
C
C     FIGURE 4.3.2.1-37A DATCOM                    *******
      XACBW  = 0.5 + A1*BDOC -.1057*BDOC**2 +.0172*BDOC**3
```

```
      IF( SPEED.GE.1.2 ) GO TO 290                                      R1T46301
C                                                                       R1T46302
      XACN    = XACN1 + (XACN -XACN1) *(SPEED-FMCRO)/(1.2-FMCRO)        R1T46303
      XACBW   = XACBW1+ (XACBW-XACBW1)*(SPEED-FMCRO)/(1.2-FMCRO)        R1T46304
C                                                                       R1T4631
  290 CLAB    = C1(55)                                                  R1T4632
      CLAW    = C1(54)                                                  R1T4633
      FKBW    = 0.0004 +1.2662*DOB +.6018*DOB**2 +.1263*DOB**3          R1T4634
      FKWB    = 1.0028 +.7116*DOB +.42*DOB**2 -.1366*DOB**3             R1T4635
      CLABW   = FKBW/(FKBW + FKWB) * CLAW                               R1T4636
      CLAWB   = FKWB/(FKBW + FKWB) * CLAW                               R1T4637
C                                                                       R1T4638
  300 XACR    = (XACN * CLAB + XACWB * CLAWB + XACBW * CLABW)/          R1T4639
     1              (CLAB + CLAW)                                       R1T4640
C                                                                       R1T4641
      XACS    = (XACN * CLAB + XACSW * CLAWB + XACBW * CLABW)/          R1T4642
     1              (CLAB + CLAW)                                       R1T4643
      C1(88)  = XACS                                                    R1T4644
C                                                                       R1T4645
      IF( KPRINT(18).EQ.0 ) GO TO 400                                   R1T4646
      WRITE(6,1000) XACR, XACN,CLAB, XACWB,CLAWB, XACBW,CLABW,          R1T4647
     1              XACS, SPEED, XLEQ, FDOB                             R1T4648
C                                                                       R1T4649
  400 RETURN                                                            R1T4650
 1000 FORMAT(5X,*WBAC DUMP* /(5X,6F15.5) )                              R1T4651
      END                                                               R1T4652
```

CC = 00138

```
      SUBROUTINE ACCR( SPEED,AR,SWPLE,SWPMC,TR,SPLAN,TOC,TW,FMCRO,
     1                     XACR, CLAX, XACS )
C
C     COMPUTES AERODYNAMIC CENTER OF SINGLE PANEL WINGS
C
      COMMON /BLKBD3/ BOT(12), ATSW(6), TPRT(6), FXAC(12,6,6)
C
      COMMON /BLKBD6/ XBD6(4), YBD6(4), ZBD6(6), FBD61(4,4,6),
     1                FBD62(4,4,6), FBD63(4,4,6)
C
      COMMON /BLKCLA/ X(11)
      COMMON /BLKPRT/ KPRINT(50)
      COMMON /BLKC01/ C1(100)
      COMMON /BLKMAP/ MAP, TRANS, DY, AMAP(22), BMAP(11)
C
      DIMENSION XDY(6), YAC(6), XVAL(3), VAL(3)
      DATA  XVAL / 0.0,  0.2,  0.5 /
      DATA  XDY / .2, .4, .6, .8, 1.2, 1.6 /,
     1      YAC / .67,.585,.55,.555,.59,.61 /
C
      NI     = TW
      DY     = AMAP(NI) * TOC
      ARLO   = C1(71)
C
      SB     = 2.
      IF( SPEED.EQ.1.0 ) GO TO 20
      IF( SPEED.GT.1.0 ) GO TO 10
C
      TANOB  = TAN(SWPLE)/SQRT(1.-SPEED**2)
      IF( TANOB.LE.1.0 ) SB = TANOB
      IF( TANOB.GT.1.0 ) SB = 2. - 1./TANOB
      GO TO 20
C
   10 TANOB  = TAN(SWPLE)/SQRT(SPEED**2 -1.)
      IF( TANOB.LE.1. ) SB = 4.-TANOB
      IF( TANOB.GT.1. ) SB = 2.+1./TANOB
C
   20 ARTSW  = AR * TAN(SWPLE)
C
      CALL TLNT(SB,ARTSW,TR, XAC1, BOT,ATSW,TPRT,FXAC, 12,6,6, 12,6)
C
      KPR11       = KPRINT(11)
      KPRINT(11) = 0
      X(1)    = SPLAN
      X(2)    = 0.0
      X(3)    = TR
      X(4)    = AR
      X(5)    = 0.0
      IF( TW.EQ.8. ) X(5) = 0.0334
      X(6)    = 0.0
      X(7)    = 0.0
      IF( TW.EQ.8. ) X(7) = 0.09
      X(8)    = 0.0
      X(9)    = 0.0
      IF( TW.EQ.8. ) X(9) = 1.173763 * TOC
      X(10)   = SWPMC
```

```
   30 X(11)  = SPLAN
C
      CALL AER2(SPEED, CLAX)
C
      XAC2   = XAC1
      VBAR   = (SPEED**2 -1.)/TOC**0.6667
      ABAR   = AR * TOC**0.3333
C
      IF( VBAR.GE.1.0.OR.VBAR.LE.-2. ) GO TO 40
C
C     TRANSONIC AERODYNAMIC CENTER
C
      CALL TLNT(ABAR,VBAR,ARTSW,VAL(1),XBD6,YBD6,ZBD6,FBD61,4,4,6,4,4)
      CALL TLNT(ABAR,VBAR,ARTSW,VAL(2),XBD6,YBD6,ZBD6,FBD62,4,4,6,4,4)
      CALL TLNT(ABAR,VBAR,ARTSW,VAL(3),XBD6,YBD6,ZBD6,FBD63,4,4,6,4,4)
C
      CALL LNTP(TR, XAC2, XVAL, VAL, 3, 2)
C
   40 XACR   = XAC1
      IF( SPEED.LE.FMCRO.OR.SPEED.GE.1.2 ) GO TO 50
      FM1    = FMCRO + 0.05
      FM2    = SQRT(1.+ TOC**.66667)
      XACR   = XAC2
C
      IF( SPEED.GT.FM1.AND.SPEED.LT.FM2 ) GO TO 50
      IF( SPEED.LE.FM1 ) XACR = XAC1 + (XAC2-XAC1)*(SPEED-FMCRO)/.05
      IF( SPEED.GE.FM2 ) XACR = XAC2 + (XAC1-XAC2)*(SPEED-FM2)/
     1                    (1.2 - FM2)
C
   50 ARHI   = 2.* ARLO
      XACS   = (1.+2.*TR)/12.* ARTSW  + (1. + TR**2/(1.+TR))/3.
      IF( AR.GT.ARHI ) GO TO 60
      CALL LNTP(DY, XACS1, XDY, YAC, 6, 2)
      XACS   = XACS1 + (1.+2.2*TR)/17.544 * ARTSW -0.2
C
   60 IF( KPRINT(18).EQ.2 ) WRITE(6,1000) SPEED, AR, SWPLE, SWPMC,
     1                      TR, SPLAN, TOC, TW, FMCRO, XACR, CLAX,
     2                      SB, ARTSW, XAC1, ABAR,  VBAR, VAL, XAC2,
     3                      FM1, FM2, ARHI, XACS
C
 1000 FORMAT(5X, *ACCR DUMP* /(5X,6F15.5) )
C
      KPRINT(11) = KPR11
      RETURN
      END
```

CC = 00101

```
      OVERLAY(3,4)
      PROGRAM LSHL
C
      COMMON /BLKHLS/ NHLSV, DFI(3,5), CPFI(5), DSI(5), CPSI(5),
     1                DELCD(5), H(5), DFI2(3,5), CPF2(5), DSI2(5),
     2                CPS2(5)
      COMMON /BLKA04/ IHLS,NXF,NXS, A2(62), BF1I, BF1O, CF1(3), BF2I,
     1                BF2O, CF2(3), BS1I, BS1O, CS1, BS2I, BS2O, CS2
      COMMON /BLKC01/ C1(85), CPF, CPS, XACS, DCDLG, C2(11)
      COMMON /BLKG01/ G1(200)
      COMMON /BLKC02/ CO2(11)
C
      COMMON /BLKOV3/ ITYP, OV3A(7), JPASS, ROT, OTE, ITRIM
C
      DIMENSION DF(3), CFOC(3), COL(10)
      DATA  DF / 3*0.0 /, CFOC / 3*0.0 /
      DATA  COL / 10*0.0 /
C
      DO 700 I = 1, NHLSV
      DF(1) = DFI(1,I)
      DF(2)  = DFI(2,I)
      DF(3)  = DFI(3,I)
      DS     = DSI(I)
      CPF    = CPFI(I)
      CPS    = CPSI(I)
      IF( CPF.EQ.0. ) CPF = 1.
      CFOC(1)= CF1(1) / CPF
      CFOC(2)= CF1(2) / CPF
      CFOC(3)= CF1(3) / CPF
      IF( CPS.EQ.0. ) CPS = 1.
      CSOC   = CS1 / CPS
      DCDLG  = DELCD(I)
      HGT    = H(I)
      DFLAP  = DF(1) + DF(2) + DF(3)
C
      BFI    = BF1I
      BFO    = BF1O
      CF     = CF1(1) + CF1(2) + CF1(3)
      BSI    = BS1I
      BSO    = BS1O
C
      CALL SSET(DF, DS, CFOC, CSOC, OTE, ROT)
      CALL DSET(BFI, BFO, BSI, BSO, CF)
      CALL MSET(BFI, BFO, BSI, BSO)
      IF( IHLS.EQ.1 ) GO TO 500
C
C     CALCULATION OF THE EFFECT OF THE SECOND HIGH LIFT SEGMENT
C
      DF(1)  = DFI2(1,I)
      DF(2)  = DFI2(2,I)
      DF(3)  = DFI2(3,I)
      DS     = DSI2(I)
      CPF    = CPF2(I)
      CPS    = CPS2(I)
      CFOC(1)= CF2(1) * CPF
      CFOC(2)= CF2(2) * CPF
```

```
      CFOC(3)= CF2(3) * CPF
      CSOC    = CS2 * CPS
C
      DA      = CO2(8) - G1(82)
      DR      = CO2(11) - 1.
      DO 300 J = 1, 10
  300 COL(J) = CO2(J)
C
      BFI     = BF2I
      BFO     = BF2O
      CF      = CF2(1) + CF2(2) + CF2(3)
      BSI     = BS2I
      BSO     = BS2O
C
      CALL SSET(DF, DS, CFOC, CSOC, OTE, ROT)
      CALL DSET(BFI, BFO, BSI, BSO, CF)
      CALL MSET(BFI, BFO, BSI, BSO )
C
      CO2(8) = CO2(8) + DA
      CO2(11)= CO2(11) + DR
      DO 400 J = 1, 10
  400 IF( J.NE.8 ) CO2(J) = CO2(J) + COL(J)
C                                               L(I)
  500 ALPHA   = -1.
      CL      = - 10.
  600 ALPHA   = ALPHA + 1.
      CL1     = CL
      CALL DAERO(ALPHA, CL, CD, CM, AGRD, CLG, CDG, CMG, HGT, DFLAP)
      IF( ALPHA.EQ.0.0 ) WRITE(6,2000)
      IF( HGT.EQ.0.0 ) WRITE(6,2001) ALPHA, CL, CD, CM
      IF( HGT.GT.0.0 ) WRITE(6,2001) ALPHA,CL,CD,CM,AGRD,CLG,CDG,CMG
C
      IF( ALPHA.GT.60.0 ) GO TO 700
      IF( CL.GT.CL1 ) GO TO 600
  700 CONTINUE
 2000 FORMAT(1H1, 20X, *HIGH LIFT SURVEY* // 15X, *ALPHA*  10X,
     1   * CL * 10X,* CD * 10X,* CM *10X,*AGRD*10X,* CLG*10X,* CDG*
     2        10X, *CMG * / )
 2001 FORMAT(5X, F15.2, 3F15.5, F15.3, 3F15.5)
      END
```

CC = 00096

```
      SUBROUTINE SSET(DF, DS, CFOC, CSOC, DTE, ROT)
C
C     COMPUTES SECTION INCREMENTS FOR FLAPS AND SLATS
C
      COMMON /BLKPRT/ KPRINT(50)
C
      COMMON /BLKC01/ C1(77), DC1F, DC1MF, DCDFS, DCMFS, DC1S,
     1                DC1MS, DCDSS, DCMSS, CPF, CPS, C2(13)
C
      COMMON /BLKA04/ IHLS, NXF, NXS, TF, TS, XDF(5), FCPF(5),
     1                FDC1F(5), FDCMXF(5), FCDF(5), FCMF(5),
     2                XDS(5), FCPS(5), FDC1S(5), FDCMS(5), FDCDS(5),
     3                FDMS(5), A04(16)
C
      DIMENSION DF(3),CFOC(3), ETA(3), C10(3), DCL(3), RMAX(3),
     1          DCLM(3), PCM(3), DCM(3), X1(4), Y1(8), F1(4,8),
     2          X2(5), Y2(6), F2(5,6), X3(6), Y3(4), F3(6,4),
     3          A0(4), A1(4), XROT(5), YEMAX(5,3),XXDS(4), YED(4,3)
C
      DATA  A0  / 0.915, 1.215, 1.39, 1.545 /,
     1      A1  / -.01186, -.0115, -.0125, -.0130 /,
     2      X1  / 0., .1, .2, .3 /,
     3      Y1  / 0., 10., 20., 30., 40., 50., 60., 70. /,
     4      F1  / 5*0.0,.001,.004,.0085, 0.,.009,.02,.032,
     5            0.,.019,.0425,.07, 0.,.0325,.071,.118,
     5            0.,.049,.104,.17, 0.,.063,.14,.237,
     5            0.,.084,.18,.295 /,
     6      X2  / 0., .1, .2, .3, .4 /,
     7      Y2  / 0., 10., 20., 30., 40., 50. /,
     8      F2  / 6*0.0,.001,.003,.006,.012, 0.,.003,.0075,.013,.023,
     9            0.,.011,.024,.0385,.059, 0.,.02,.045,.079,.117,
     9            0.,.03,.069,.127,.205  /,
     1      X3  / 0., 1., 2., 4., 6., 8. /,
     2      Y3  / .2, .3, .4, .656 /,
     3      F3  / 0.,.01,.02,.058,.155,.28,0.,.01,.02,.05,.12,.24,
     4            0.,.01,.02,.046,.091,.2,0.,.008,.016,.032,.057,.106/
      DATA  XROT / 0.0, 0.07, 0.08, 0.1, 0.2 /,
     1     YEMAX / 0.56, 0.98, 1.02, 0.95, 0.0,
     2             .56,1.34,1.74,1.72,.8,.56,1.09,1.2,1.15,.2 /,
     3      XXDS / 0.0, 15.0, 25.0, 45.0 /,
     4       YED / 1.0, 1.0, .96, .25,  1.0,1.0,.75,0.,
     5             1.0, 1.0, .97, .67  /
C
       DC1F   = 0.0
       DC1MF  = 0.0
       DCDFS  = 0.0
       DCMFS  = 0.0
       DC1S   = 0.0
       DC1MS  = 0.0
       DCDSS  = 0.0
       DCMSS  = 0.0
C
       IF( TF.EQ.0. ) GO TO 100
       IF( TF.GE.5. ) GO TO 60
C
C      TRAILING-EDGE DEVICE              ****
```

```
      NF     = TF
      CFP    = CFOC(1) + CFOC(2) + CFOC(3)
      IF (TF.EQ.1.)  ZKM = 1.0031 -.928*CFP + 9.9869*CFP**2
     1              -7.1843*CFP**3
      IF( TF.EQ.1.0.AND.CFP.LT.0.1 ) ZKM = 1.0
      IF( TF.GT.1.) ZKM = 1.2029 -1.1834*CFP +13.145*CFP**2
     1              -11.5455*CFP**3
      IF( TF.GT.1.0.AND.CFP.LT.0.1 ) ZKM = 1.2
      ZKD    = 1.9 - 0.0225 * (DF(1)+DF(2)+DF(3))
      IF( ZKD.LT.1.0 ) ZKD = 1.0
      ZKT    = 1.23 + 4.5 * ROT
      ZKMAX  = ZKT * ZKD
C
      IF( TF.LE.2. ) NT = 1
      IF( TF.EQ.3. ) NT = 2
      IF( TF.EQ.4. ) NT = 3
C
      PHI    = DF(1)
C
      DO 50 I = 1, NT
      IF( I.GT.1 ) PHI = DF(I) + PHI
      IF( I.EQ.NT) PHI = PHI + DTE
      IF( I.GT.1 ) CFP = CFP - CFOC(I-1)
      ETA(I) = A0(NF) + A1(NF)*PHI
      IF(TF.EQ.1.0.AND.PHI.GT.35.0 ) ETA(I) = 1.5912 -.05068*PHI
     1      +.000754*PHI**2 -.0000039*PHI**3
      IF( ETA(I).GT.0.77 ) ETA(I) = 0.77
C
      DFR    = DF(I)/57.2956
      OF     = ACOS(1. - 2.*CFP)
      C10(I) = 2.*(OF + SIN(OF))
C
      DCL(I) = ETA(I) * C10(I) * DFR
      DC1F   = DC1F + DCL(I)
C
      XS     = 0.0
      IF( TS.GT.0.0.AND.DS.GT.0.0 ) XS = CSOC
      X      = ACOS(2.*XS - 1.)
      RMAX(I) = 1. -(OF/(OF + SIN(OF))) * (1.+ ALOG(ABS(SIN(.5*(X+OF))/
     1                SIN(.5*(X-OF)) ) )/(OF * TAN(.5*X) ) )
C
      DCLM(I) = DCL(I) * RMAX(I) * ZKMAX
      IF( DCLM(I).GT.DCL(I) ) DCLM(I) = DCL(I)
      DC1MF   = DC1MF + DCLM(I)
C
      RCM(I) = - CFP   *  SIN(OF)/C10(I)
C
      DCM(I) = DCL(I) * RCM(I) * ZKM
      DCMFS  = DCMFS + DCM(I)
C
   50 CONTINUE
C
      DCL2   = DC1F*DC1F
      CFP    = CFOC(1) + CFOC(2) + CFOC(3)
      IF( TF.EQ.1. ) DCDFS = DLNT(CFP, DF(1), X1, Y1, F1, 4,8,4,4,2)
      IF( TF.EQ.2. ) DCDFS = DLNT(CFP, DF(1), X2, Y2, F2, 5,6,5,4,2)
```

```
      IF( TF.GF.3.) DCDFS = DLNT(DCL2, CFP, X3, Y3, F3, 6,4, 6,4,2)
C
      GO TO 100
   60 CALL LNTP(DF(1), DC1F, XDF, FDC1F, NXF, 4)
      CALL LNTP(DF(1), DC1MF,XDF, FDCMXF,NXF, 4)
      CALL LNTP(DF(1), DCDFS,XDF, FCDF,  NXF, 4)
      CALL LNTP(DF(1), DCMFS,XDF, FCMF,  NXF, 4)
      CALL LNTP(DF(1), CPF,  XDF, FCPF,  NXF, 4)
C
  100 IF( TS.LE.0. ) GO TO 200
      IF( TS.GE.4. ) GO TO 150
C     LEADINE-EDGE DEVICE                 ****
C
      DS     = ACOS(1. - 2.*CSOC)
      C1LE   = 2.*(SIN(DS) - DS)
      CMXLE  = 2.* SIN(DS)
      CMLE   = (1.-CSOC)*SIN(DS) ’C1LE
      DSR    = DS/57.2956
      NS     = TS
      CALL LNTP(ROT, EMAX, XROT, YEMAX(1,NS), 5,2)
      CALL LNTP(DS, ED,XXDS, YED(1,NS), 4, 2)
C
      DC1S   = C1LE * DSR
      DC1MS  = CMXLE * EMAX * ED * DSR
      DCDSS  = 0.0
      DCMSS  = CMLE * DC1S
C
      GO TO 200
  150 CALL LNTP(DS, DC1S, XDS, FDC1S, NXS, 4)
      CALL LNTP(DS, DC1MS,XDS, FDCMS, NXS, 4)
      CALL LNTP(DS, DCDSS,XDS, FDCDS, NXS, 4)
      CALL LNTP(DS, DCMSS,XDS, FDMS,  NXS, 4)
      CALL LNTP(DS, CPS,  XDS, FCPS,  NXS, 4)
C
  200 IF( KPRINT(27).EQ.1 ) WRITE(6,1000) DC1F, DC1MF, DCDFS, DCMFS,
     1          DC1S, DC1MS, DCDSS, DCMSS, CPF, CPS, DCL, ETA, C10,
     2          DCLM, RMAX, ZKMAX, DCM, RCM, ZKM, C1LE, CMXLE, EMAX,
     3          ED, TF, TS, DF, DS, CFOC, CSOC, OTE, ROT
C
 1000 FORMAT(5X, *SSET DUMP* /(5X, 7F15.5) )
C
      RETURN
      END
```

CC = 00155

```
      SUBROUTINE DSET(BFI, BFO, BSI, BSO, CF)                             R1T5009
C                                                                         R1T5010
C     AERODYNAMIC CONTROL ROUTINE FOR HIGH LIFT SYSTEMS                   R1T5011
C                                                                         R1T5012
      COMMON /BLKC02/ DCLOF, DCLOS, DCLMF, DCLMS, DCDMIN, DCLF, DCLS,     R1T5013
     1                SPLANX, DCMO, DCMCL, RCLA                           R1T5014
      COMMON /BLKC01/ C1(11), AK, DCL, C2(64), DC1F, DC1MF, DCDFS,        R1T5015
     1                DCMFS,DC1OS,DC1MS,DCDSS,DCMSS,CPF,CPS,C3(13)        R1T5016
      COMMON /BLKG01/ G1(200)                                             R1T5017
C                                                                         R1T5018
      COMMON /BLKMX3/ SPAN(6),  YTPR(4), FKB( 6,4), FKS(6),  FKD( 6,4)   R1T5019
      COMMON /BLKMX9/ SPAN4(5),YBFI1(5), FKA(5,5),                        R1T5020
     1                SPAN5(5), YBFI2(3), FKF(5,3)                        R1T5021
C                                                                         R1T5022
      ZKT    = 1.0                                                        R1T5023
      SWPLE  = G1(103)                                                    R1T5024
      SWPMC  = G1(49)                                                     R1T5025
      TPR    = G1(81)                                                     R1T5026
      AR     = G1(80)                                                     R1T5027
      SPLAN  = G1(82)                                                     R1T5028
      DCLS   = 0.0                                                        R1T5029
C                                                                         R1T5030
      ZKBI   = DLNT(BFI, TPR, SPAN, YTPR, FKB,  6,4, 6,2,2)               R1T5031
      ZKBO   = DLNT(BFO, TPR, SPAN, YTPR, FKB,  6,4, 6,2,2)               R1T5032
      ZKB    = ZKBO - ZKBI                                                R1T5033
C                                                                         R1T5034
      ASC    = SQRT(1. - (1.-CF)**1.61 )                                  R1T5035
      ZKC    = 1. + 1.875 * (1.-ASC)**2 /AR                               R1T5036
C                                                                         R1T5037
      CALL LNTP(BSI, ZKSI, SPAN, FKS,  6, 4)                              R1T5038
      CALL LNTP(BSO, ZKSO, SPAN, FKS,  6, 4)                              R1T5039
      ZKS    = ZKSO - ZKSI                                                R1T5040
C                                                                         R1T5041
C                                                                         R1T5042
      SPLANX = SPLAN * (1. + (CPF-1.)/(1.+TPR) * (2.-(1.-TPR)             R1T5043
     1              *(BFO + BFI)) *(BFO - BFI)                            R1T5044
     2              + (CPS-1.)/(1.+TPR) * (2.-(1.-TPR)*(BSO-BSI))         R1T5045
     3              * (BSO - BSI) )                                       R1T5046
C                                                                         R1T5047
   20 CLOC1  = AR/(2. + SQRT(4.+ (SPLAN * AR/SPLANX /COS(SWPMC))**2) )    R1T5048
      CLOC1R = AR/(2.+ SQRT(4.+ (AR/COS(SWPMC))**2 ) )                    R1T5049
      RCLA   = CLOC1/CLOC1R                                               R1T5050
C                                                                         R1T5051
      DCLOF  = ZKT * DC1F * CLOC1 * ZKC * ZKB                             R1T5052
      DCLMF  = ZKT * DC1MF * CLOC1 * ZKC * ZKB * COS(SWPMC)               R1T5053
C                                                                         R1T5054
      DCLOS  = DC1OS * CLOC1 * ZKS                                        R1T5055
      DCLMS  = DC1MS * CLOC1 * ZKS * COS(SWPLE)                           R1T5056
C                                                                         R1T5057
C                                                                         R1T5058
   30 WRITE(6,1000) ZKB, ZKC, ZKS, CLOC1, DC1F, DC1MF, DC1OS,             R1T5059
     1                DC1MS, DCLOS, DCLOF                                 R1T5060
C                                                                         R1T5061
C     CALCULATION OF DRAG PARAMETERS                                      R1T5062
C                                                                         R1T5063
      SWPHL  = ATAN(TAN(SWPLE) - 4./AR * (1.-TPR)/(1.+TPR) * (1.-CF) )    R1T5064
```

```
C
      ZKDI    = DLNT(BFI, TPR, SPAN, YTPR, FKD,  6, 4,  6, 2,2)
      ZKDO    = DLNT(BFO, TPR, SPAN, YTPR, FKD,  6, 4,  6, 2,2)
      ZKD     = ZKDO - ZKDI
C
   40 CONTINUE
      E       = AR/6.283
      D       = BFO
C
      ZKA = DLNT(E, BFI,SPAN4, YBFI1,FKA, 5, 5, 5, 4, 2 )
      ZKF = DLNT(D, BFI,SPAN5, YBFI2,FKF, 5, 3, 5, 2, 2 )
C
C
C
   45 WRITE(6,1001) SWPHL, ZKD, ZKA, ZKF, CDO, SWPLE, AR, TPR, AK, DCL
      EK      = 0.31831/AR/AK
C
   50 CONTINUE
      DCLF    = (1.-EK) * DCLOF/(1.+1.16*CLOC1*(.5-CF) )
      CDC     = 0.0
      IF( EK.LT.1.0 ) CDC = (DCLF**2 + 2.*DCLF*DCL)*0.31831/
     1                      (AR * (1.-EK))
C
      DCDF    = DCDFS * COS(SWPHL) * ZKD
      DCDS    = DCDSS * COS(SWPLE) * ZKD
      CDI     = ZKA * ZKF * DCLOF**2 *0.31831/AR
C
      DCDMIN = DCDF + DCDS + CDI + CDC
      WRITE(6,1002) DCDMIN, CDI, DCDS, DCDF, DCL, DCLF, CDC
C
   60 RETURN
 1000 FORMAT(1H1, 10X, *HIGH LIFT CONSTANTS*, // 10X
     1       *ZKB  =* F10.6, 10X *ZKC  =* F10.6, 10X *ZKS  =*
     2       F10.6 / 10X  *CLOC1=* F10.6, 10X *DC1F =* F10.6,
     3       10X   *DC1MF=* F10.6  / 10X  *DC1OS=* F10.6, 10X
     4       *DC1MS =* F10.6, 10X
     6       *DCLOS=* F10.6 /10X *DCLOF=* F10.6 )
 1001 FORMAT( /// 10X  *SWPHL=* F10.6, 10X *ZKD  =*  F10.6,
     1       10X *ZKA  =* F10.6 / 10X  *ZKF  =* F10.6, 10X
     2       *CDO  =* F10.6, 10X *SWPLE=* F10.6 / 10X
     3       *AR   =* F10.6, 10X *TPR  =* F10.6,  10X
     4       *AK   =* F10.6 /10X *DCL  =* F10.6 )
 1002 FORMAT( ///10X, *DCDMIN =* F10.6, 10X *CDI =* F10.6, 10X
     1       *DCDS =* F10.6 /10X *DCDF =* F10.6, 10X
     2       *DCL =* F10.6, 10X *DCLF =* F10.6 /10X *CDC =*F10.6 )
 1003 FORMAT (6F10.6)
      END
```

CC = 00103

```
      SUBROUTINE MSET(BFI, BFO, BSI, BSO )
C
C     COMPUTES MOMENT SHIFT DUE TO FLAPS AND SLATS AT CL = 0
C
      COMMON /BLKC02/ DCLOF, DCLGS, DCLMF, DCLMS, DCDMIN, DCLF, DCLS,
     1                SPLANX, DCMO, DCMCL, RCLA
      COMMON /BLKC01/ C1(77), DC1F, DC1MF, DCDFS, DCMFS,
     1                DC1S, DC1MS, DCDSS, DCMSS, CPF, CPS, C2(13)
      COMMON /BLKG01/ G1(200)
C
      COMMON/BLKMX7/ SPAN2(11), YTPR2(5), FKSW(11,5)
      COMMON/BLKMX8/ SPAN3(6), YTPR3(4), FKM(6,4)
C
      TPR    = G1(81)
      AR     = G1(80)
      SWPMC  = G1(49)
      CMCL   = C1(15)
      CLA    = C1(17)
      ALO    = C1(18)
C
C     FLAP CALCULATIONS START HERE
C
      DCMAF  = 0.0
      DCMCLF = 0.0
      IF( DCMFS.EQ.0.0.AND.DC1F.EQ.0.0.AND.CPF.EQ.0.0 ) GO TO 40
C
      ZKSWI= DLNT(BFI, TPR, SPAN2, YTPR2, FKSW, 11, 5, 11, 2, 2)
      ZKSWO= DLNT(BFO, TPR, SPAN2, YTPR2, FKSW, 11, 5, 11, 2, 2)
      ZKSW = ZKSWO - ZKSWI
C
      ZKMO = DLNT (BFO, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)
      ZKMI = DLNT (BFI, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)
      ZKM = ZKMO - ZKMI
C
      DCMAF  = DCMFS * CPF**2 * ZKM  + 0.5 * AR * TAN(SWPMC)
     1                               * DC1F * ZKSW
C
      DCMCLF = (-0.25 * CPF *(CPF-1.) + CMCL * (CPF**2-1.))* ZKM
   40 CONTINUE
C
C     SLAT CALCULATIONS START HERE
C
      DCMAS  = 0.0
      DCMCLS = 0.0
      IF( DCMSS.EQ.0.0.AND.DC1S.EQ.0.0.AND.CPS.EQ.0.0 ) GO TO 50
      ZKMSO = DLNT(BSO, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)
      ZKMSI = DLNT(BSI, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)
      ZKMS  = ZKMSO - ZKMSI
C
      ZKSWSO = DLNT(BSO, TPR, SPAN2, YTPR2, FKSW, 11, 5, 11, 2, 2)
      ZKSWSI = DLNT(BSI, TPR, SPAN2, YTPR2, FKSW, 11, 5, 11, 2, 2)
      ZKSWS  = ZKSWSO - ZKSWSI
C
      DCMAS  = DCMSS * CPS**2 * ZKMS + 0.5 * AR * TAN(SWPMC)
     1                               * DC1S * ZKSWS
C
```

```
      DCMCLS = (-0.25 * CPS *(CPS-1.) + CMCL * (CPS**2-1.))* ZKMS
C
C
C     ADD FLAP AND SLAT MOMENTS
C
   50 DCMA   = DCMAF + DCMAS
      DCMCL  = DCMCLF + DCMCLS
C
      CLO    = -CLA * ALO
      DCMO   = DCMA - CMCL * (DCLOF + DCLOS) - CMCL * (CLO+DCLOF+DCLOS)
C
C
      WRITE(6,1001) DCMAF, DCMAS, DCMO
C
 1001 FORMAT(///10X,*DCMOF = *F10.6/10X,*DCMOS = *F10.6/
     1           10X, *DCMO  = * F10.6)
      RETURN
      END
```

CC = 00074

```
      SUBROUTINE DAERO(ALPHA, CL, CD, CM, AGRD, CLG, CDG, CMG, H, DF)
C
C     COMPUTES LIFT MOMENT AND DRAG FOR HIGH-LIFT SYSTEMS IN
C     FREE AIR OR GROUND EFFECT
C
      COMMON /BLKC02/ DCLOF, DCLOS, DCLMF, DCLMS, DCDMIN, DCLF, DCLS,
     1                SPLANX, DCMO, DCMCL, RCLA
      COMMON /BLKG01/ G1(200)
      COMMON /BLKA02/ A1(433)
      COMMON /BLKA03/ A03(12), CMAC, A03A(7)
      COMMON /BLKA04/ A2(66), BF1C, A3(4), BF20, A4(9)
      COMMON /BLKC01/ C1(4), CDMINB, C2(6), FK, DELCL, CMO, CMCL,
     1                XACWB, CLA, ALO, C3(19), CDWING(4), C7(12),
     1                CLAW, CLAN, CLAT, AH, BH, HSTAR,
     2                C4(5), CLMAXB, ABRK, AMAX, DAMAX,
     3                C5(25), DEDA, C6(6)
C
      DIMENSION XDX(6), YA(6), YB(6), XTR(6), YAR(4), Z1(6,4),
     1          XBF(6), Z2(6)
C
      DATA  XDX / -1., -.6, -.2,  .2, .6, 1. /,
     1      YA  / -3.71,-3.03,-2.63,-2.34,-2.13,-2. /,
     1      YB  / .466,.5,.664,.901,1.1,1.171 /
C
C     FIG.S  4.7.1-18A&B IN DATCOM
      DATA  XTR / 1., 2., 3., 4., 5., 6. /,
     1      YAR / 4., 6., 8., 10. /,
     2      Z1  / .825,.772,.745,.726,.715,.712, .855,.772,.738,.715,
     3            .7,.695, .88,.775,.727,.702,.69,.685,
     4            .895,.776,.723,.697,.68,.675 /,
     5      XBF / 0., .2, .4, .6, .8, 1. /,
     6      Z2  / 0., .34, .58, .76, .89, 1. /
C
      SPLAN  = G1(82)
      AR     = G1(80)
      TR     = G1(81)
      DCDLG  = C5(21)
      ESWOC  = G1(102)
      OMEGA  = G1(110)
      XH     = G1(109)
C
      CLO    = -CLA*ALO + DCLOF + DCLOS
      CLMAX  = CLMAXB + DCLMF + DCLMS
      CLAHL  =  CLAN + CLAT + CLAW * RCLA
      AK     = FK * SPLAN/SPLANX
      DCL    = DELCL + DCLF + DCLS
      CDMIN  = CDMINB + DCDMIN + DCDLG
C
      ALIN   = (CLMAX - CLO)/CLAHL  - DAMAX
C
      CL     = CLAHL * ALPHA + CLO
      IF( ALPHA.LE.ALIN ) GO TO 20
C
      DELM   = CLAHL * (ALIN + 2.* DAMAX) + CLO - CLMAX
      DEL    =((ALPHA - ALIN) *0.5/DAMAX )**2 * DELM
C
```

```
      CL      = CL - DEL
C
   20 CD      = CDMIN +AK*(CL - DCL)**2
      IF( ALPHA.LE.ALIN ) GO TO 30
C
      EO      = 0.31831/AR/AK
      E       = EO + (0.3 - EO) * 0.5 * (ALPHA - ALIN)/DAMAX
      IF( E.LE.0.0 ) E = 0.1
      CD      = CDMIN + 0.31831/AR/E * (CL - DCL)**2
C
   30 CMOTOT  = CMO + DCMO
      DCMDCL  = CMCL + DCMCL
      CLTAIL  = CLAT * (ALPHA-HSTAR)
      CLWB    = CL - CLTAIL
      CM      = CMOTOT + DCMDCL * CLWB
     1          -CLTAIL * XH * COS(OMEGA - ALPHA/57.3)/CMAC
C
C     GROUND EFFECTS ARE NOW CALCULATED
C
      DCLWG   = 0.0
      DCLTG   = 0.0
      DCMWG   = 0.0
      DCMTG   = 0.0
      DCDG    = 0.0
      AGRD    = ALPHA
      OMEGA   = G1(110)
      BO2     = A1(240)
      DXOB    = G1(106)
      CROB    = G1(105)
      HTZ     = A1(432)
C
      IF( H.EQ.0.0 ) GO TO 50
      CDWNG   = CDWING(1) + CDWING(2) + CDWING(3) + CD - CDMIN
      CALL LNTP(DXOB, A, XDX, YA, 6, 4)
      CALL LNTP(DXOB, B, XDX, YB, 6, 4)
      SIGP    = 2.71828 **(A * (H/BO2)**B )
C
      HOC     = H/(CROB*BO2) **0.5
      A2D     = 9.1189 * CL /(COS(ESWQC)**2)
      DLOLO   = 0.058/(HOC**1.7)  -0.0085/HOC * A2D
      IF( HOC.LT.0.4 ) DLOLO = DLOLO - 0.02/(HOC**2) + 0.05/HOC
C
      DDCLF   = -.142 +.2272*HOC -.0902*HOC**2
C
      SIG     = 2.71828 **(-2.48 * (H/BO2)**.768)
C
      R       = SQRT(1. +(H/BO2)**2 ) - H/BO2
      T       = 0.03979 * HOC /(HOC**2 - .015625)
      DCLWG   = (9.12/AR +7.16*CROB) * CLWB * CLAW * SIGP * RCLA
     1             + 0.5 * AR * CROB * DLOLO * R * CLWB
     2           + DDCLF * (DF/50.)**2
C
      DCDG1   = -SIG * 0.31831*CLWB**2/AR
      DCDG    = DCDG1 -(CDWNG + DCDG1) *R *T *CLWB
C
C     GROUND EFFECTS ON HORIZONTAL TAIL ARE CALCULATED
```

```
      IF( CLAT.EQ.0.0 ) GO TO 40
C
      TRIV    = 6.
      IF( TR.NE.0.0 ) TRIV = 1./TR
      BPWOB   = DLNT(TRIV, AR, XTR, YAR, Z1, 6,4, 6, 4,2)
C
      BF      = BF10
      IF( BF20.GT.BF ) BF = BF20
      CALL LNTP(BF, BFOBWP, XBF, Z2, 6, 4)
C
      SPAN    = BO2 *2.
      BWP     = BPWOB * SPAN
      BFP     = BWP * BFOBWP
C
      BEFF    = BWP
      IF( BFP.NE.0.0 ) BEFF = CLO/((CLO -DCLOF)/BWP + DCLOF/BFP)
C
      HH      = (H + HTZ)*COS(ALPHA/57.3) - XH*SIN(ALPHA/57.3)
      DE      = DEDA * ALPHA * (BEFF**2 + (HH-H)**2)/(BEFF**2 +(HH+H)**2)
C
      DCLTG   = DE * CLAT/(1.-DEDA)
   40 AGRD    = ALPHA -(DCLWG +DCLTG)/CLAHL
C
      DCMWG   = DCMDCL * DCLWG
      DCMTG   = -DCLTG * XH * COS(OMEGA -AGRD/57.2956)
C
   50 CLG     = CLWB + CLAT *(AGRD-HSTAR) + DCLTG
      CMG     = CMOTOT  + DCMDCL * CLWB
     1          -(CLG-CLWB)* XH *COS(OMEGA -AGRD/57.2956)
      CDG     = CD + DCDG
C
      RETURN
      END
```

CC = 00145

```
      SUBROUTINE ATMOS (ZFT, TM, SIGMA, RHO, THETA, DELTA, CA, AMU, K)    CS
C   THIS IS A SUBROUTINE TO COMPUTE CERTAIN ELEMENTS OF THE 1962          CS
C   U.S. STANDARD ATMOSPHERE UP TO 90 KILOMETERS.                         CS
C                                                                         CS
C     *************************************************************      CS
C                                                                         CS
C          THIS MODEL REPRESENTS MEAN ANNUAL, MID-LATITUDE, DRY AIR       CS
C     CONDITIONS.  DATA ON LATITUDINAL AND SEASONAL VARIATIONS ARE        CS
C     AVAILABLE FROM THE OFFICE OF THE STAFF METEOROLOGIST (ASD/WE).      CS
C                                                                         CS
C          CARE MUST BE EXERCISED WHEN USING THIS AND OTHER MODELS FOR    CS
C     ENGINEERING PURPOSES AND IT IS STRONGLY URGED THAT USERS CONTACT    CS1
C     ASD/WE FOR INTERPRETATION AND ASSISTANCE.                           CS1
C                                                                         CS1
C     ************************   *********************************      CS1
C                                                                         CS1
C    CALLING SEQUENCE...                                                  CS1
C                                                                         CS1
C    CALL ATMOS (ZFT, TM, SIGMA, RHO, THETA, DELTA, CA, AMU, K)           CS1
C         ZFT   =   GEOMETRIC ALTITUDE (FEET)                             CS1
C         TM    =   MOLECULAR SCALE TEMPERATURE (DEGREES RANKINE)         CS1
C         SIGMA =   RATIO OF DENSITY TO THAT AT SEA LEVEL                 CS1
C         RHO   =   DENSITY (LB-SEC**2*FT**(-4)    OR    SLUGS-FT**3)     CS1
C         THETA =   RATIO OF TEMPERATURE TO THAT AT SEA LEVEL             CS1
C         DELTA =   RATIO OF PRESSURE TO THAT AT SEA LEVEL                CS1
C         CA    =   SPEED OF SOUND (FT/SEC)                               CS1
C         AMU   =   VISCOSITY COEFFICIENT (LB-SEC/FT**2)                  CS1
C                                                                         CS1
C         K = 1 NORMAL                                                    CS1
C           = 2 ALTITUDE LESS THAN -5000 METERS OR GREATER THAN 90 KM     CS1
C           = 3 FLOATING POINT OVERFLOW                                   CS1
C                                                                         CS1
C   ALL DATA AND FUNDAMENTAL CONSTANTS ARE IN THE METRIC SYSTEM AS        CS1
C   THESE QUANTITIES ARE DEFINED AS EXACT IN THIS SYSTEM.                 CS1
C                                                                         CS1
C   THE RADIUS OF THE EARTH (REFT59) IS THE VALUE ASSOCIATED WITH THE     CS1
C   1959 ARDC ATMOSPHERE SO THAT PROGRAMS CURRENTLY USING THE LIBRARY     CS1
C   ROUTINE WILL NOT REQUIRE ALTERATION TO USE THIS ROUTINE.              CS1
      DIMENSION HB(10),TMB(10),DELTAB(10),ALM(10),ARAY(10,4)              CS1
      EQUIVALENCE(ARAY(1,1),HB),(ARAY(1,2),TMB),(ARAY(1,3),DELTAB),       CS1
     *          (ARAY(1,4),ALM)                                           CS1
      DATA ((ARAY(I,J),J=1,4),I=1,10)/                                    CS1
     X       -5.0    ,     320.65  ,    1.75363  E 00 ,    -6.5     ,     CS1
     X        0.0    ,     288.15  ,    1.00000  E 00 ,    -6.5     ,     CS1
     X       11.0    ,     216.65  ,    2.23361  E-01 ,     0.0     ,     CS1
     X       20.0    ,     216.65  ,    5.40328  E-02 ,     1.0     ,     CS1
     X       32.0    ,     228.65  ,    8.56663  E-03 ,     2.8     ,     CS1
     X       47.0    ,     270.65  ,    1.09455  E-03 ,     0.0     ,     CS1
     X       52.0    ,     270.65  ,    5.82289  E-04 ,    -2.0     ,     CS1
     X       61.0    ,     252.65  ,    1.79718  E-04 ,    -4.0     ,     CS1
     X       79.0    ,     180.6   ,    1.02410  E-05 ,     0.0     ,     CS1
     X       88.743  ,     180.65  ,    1.62230  E-06 ,     0.0     /     CS1
      DATA   REFT59/2.0855531E 07/, GZ    /9.80665/,                      CS1
     A       AMZ   /28.9644      /, RSTAR /8.31432/,                      CS1
     B       FTTOKM/3.048E-04    /, S     /110.4  /,                      CS1
```

```
     C      AMUZ  /1.2024E-05   /, CAZ   /1116.45/,                    CS1
     D      RHOZ  /0.076474     /, GZENG /32.1741/                     CS1
C  CONVERT GEOMETRIC ALTITUDE TO GEOPOTENTIAL ALTITUDE                 CS1
      HFT = (REFT59/(REFT59+ZFT))*ZFT                                  CS1
C  CONVERT HFT AND ZFT TO KILOMETERS                                   CS1
      Z = FTTOKM*ZFT                                                   CS1
      H = FTTOKM*HFT                                                   CS1
      K = 1                                                            CS1
      TMZ = TMB(2)                                                     CS1
      IF (H.LT.-5.0.OR.Z.GT.90.0)  GO TO 16                            CS1
      DO 10 M=1,10                                                     CS1
      IF (H-HB(M)) 11,12,10                                            CS1
   10 CONTINUE                                                         CS1
      GO TO 16                                                         CS1
   11 M = M-1                                                          CS1
   12 DELH = H-HB(M)                                                   CS1
      IF (ALM(M).EQ.0.0)  GO TO 13                                     CS1
      TMK = TMB(M)+ALM(M)*DELH                                         CS1
C  GRADIENT IS NON ZERO, PAGE 10, EQUATION I.2.10-(3)                  CS1
      DELTA = DELTAB(M)*((TMB(M)/TMK)**(GZ*AMZ/(RSTAR*ALM(M))))        CS1
      GO TO 14                                                         CS1
   13 TMK = TMB(M)                                                     CS1
C  GRADIENT IS ZERO, PAGE 10, EQUATION I.2.10-(4)                      CS1
      DELTA = DELTAB(M)*EXP(-GZ*AMZ*DELH/(RSTAR*TMB(M)))               CS1
   14 THETA = TMK/TMZ                                                  CS1
      SIGMA = DELTA/THETA                                              CS1
      ALPHA = SQRT(THETA**3)*((TMZ+S)/(TMK+S))                         CS1
C  CONVERSION TO ENGLISH UNITS                                         CS1
      TM = 1.8*TMK                                                     CS1
      RHO = RHOZ*SIGMA/GZENG                                           CS1
      CA = CAZ*SQRT(THETA)                                             CS1
      AMU = AMUZ*ALPHA/GZENG                                           CS1
C     CALL OVERFL(J)                                                   CS1
C       GO TO (15,17), J                                               CS1
C  15 K = K+2                                                          CS1
      GO TO 17                                                         CS1
   16 K = 2                                                            CS1
   17 RETURN                                                           CS1
      END                                                              CS1
```

```
      SUBROUTINE PLSQ(X,Y,N,K,C,LIST,EMAX,ERMS,EMEQ)                    CS1
C                                                                       CS1
C     PLSQ      POLYNOMIAL LEAST SQUARE CURVE FIT                       CS1
C                                                                       CS1
C                                                                       CS1
C         PLSQ WILL FIT A GIVEN SET OF DATA TO A                        CS1
C     POLYNOMIAL OF DEGREE K OF THE FORM...                             CS1
C                                                                       CS1
C         Y=C(K+1)+C(K)*X+C(K-1)*X**2+...+C(2)*X**(K-1)+C(1)*X**K       CS1
C                                                                       CS1
C     PLSQ THEN COMPUTES THE MAXIMUM ERROR AND ROOT                     CS1
C     MEAN SQUARE ERROR OBTAINED BY USING THE C                         CS1
C     COEFFICIENTS TO RE-COMPUTE Y FROM X                               CS1
C                                                                       CS1
C     USAGE...                                                          CS1
C                                                                       CS1
C     DIMENSION X(N), Y(N), C(L)                                        CS1
C         WHERE L IS K+1                                                CS1
C     CALL PLSQ(X,Y,N,K,C,LIST,EMAX,ERMS,EMEQ)                          CS1
C                                                                       CS1
C        WHERE,                                                         CS1
C                                                                       CS1
C        X  IS THE ARRAY OF N INDEPENDENT VARIABLES                     CS1
C                                                                       CS1
C        Y  IS THE ARRAY OF N DEPENDENT VARIABLES                       CS1
C                                                                       CS1
C        N  IS THE NUMBER OF INDEPENDENT(DEPENDENT)                     CS1
C           VARIABLES                                                   CS1
C                                                                       CS1
C        K  IS THE DEGREE OF THE LEAST SQUARES POLYNOMIAL               CS1
C                                                                       CS1
C        C  IS THE ARRAY OF THE COEFFICIENTS,HIGH ORDER                 CS1
C           TO LOW ORDER, OF THE LEAST SQUARES POLYNOMIAL               CS1
C                                                                       CS1
C        LIST  =0   SUPPRESSES THE ERROR ANALYSIS OUTPUT                CS1
C              =1   GIVES THE ERROR ANALYSIS OUTPUT                     CS1
C                                                                       CS1
C        EMAX  IS THE MAXIMUM ABSOLUTE ERROR OBTAINED                   CS1
C              BY USING THE COMPUTED C COEFFICIENTS TO                  CS1
C              APPROXIMATE THE DEPENDENT VARIABLE                       CS1
C                                                                       CS1
C        ERMS  IS THE ROOT MEAN SQUARE ERROR OBTAINED                   CS1
C              BY USING THE COMPUTED C COEFFICIENTS TO                  CS1
C              APPROXIMATE THE DEPENDENT VARIABLE                       CS1
C                                                                       CS1
C        EMEQ  IS THE MAXIMUM DEVIATION FROM UNITY                      CS1
C              IN THE LINEAR SYSTEM CHECK SOLUTION                      CS1
C                                                                       CS1
C                                                                       CS1
C     PLSQ CALLS SUBROUTINES FXEM AND MTXEQ                             CS1
C                                                                       CS1
C     PLSQ USES 1309 CELLS OF BLANK COMMON                              CS1
C                                                                       CS1
C                                                                       CS1
      COMMON    MTXEQT(664), CF, DIF, I, J, JC, JK,                     CS1
```

```
     *          L, LL, LU, M, SUM, XI, XM(576),                         CS1
     *          XMAX, XMIN, XP, YC, YM(48)                              CS1
      DIMENSION    X(N), Y(N), C(24),                                   CS1
     *             XDP(48), XYDP(24)                                    CS1
      EQUIVALENCE    (MTXEQT(1),XDP(1)), (MTXEQT(97),XYDP(1))           CS1
      DATA    KMAX/ 23/                                                 CS1
C                                                                       CS1
C     CHECK K AND N FOR PROPER RANGE                                    CS1
C                                                                       CS1
      IF (K .GT. KMAX .OR. N .LE. K .OR. K .LE. 0)    GO TO 200         CS1
      L=K+1                                                             CS1
C                                                                       CS1
C     FIND MINIMUM AND MAXIMUM VALUES FOR X                             CS1
C                                                                       CS1
      XMIN=X(1)                                                         CS1
      XMAX=X(1)                                                         CS1
      DO 10 I=2,N                                                       CS1
      XMIN = AMIN1(XMIN,X(I))                                           CS1
   10 XMAX = AMAX1(XMAX,X(I))                                           CS1
C                                                                       CS1
C     ZERO DOUBLE PRECISION ARRAYS FOR SUMMING                          CS1
C                                                                       CS1
      M=2*K+1                                                           CS1
      DO 20 I=1,M                                                       CS1
   20 XDP(I) = 0.0                                                      CS1
      DO 25 I=1,L                                                       CS1
   25 XYDP(I) = 0.0                                                     CS1
C                                                                       CS1
C     TRANSFORM RANGE OF X TO (-1,+1) AND                               CS1
C     COMPUTE SUMS OF POWERS OF X AND SUMS                              CS1
C     OF Y TIMES POWERS OF X                                            CS1
C                                                                       CS1
      C1 = 2.0 / (XMAX - XMIN)                                          CS1
      C2 = (XMAX + XMIN) / (XMAX - XMIN)                                CS1
      LL=K+2                                                            CS1
      LU=2*K+1                                                          CS1
      DO 40 I=1,N                                                       CS1
      XP = 1.0                                                          CS1
      XI = C1 * X(I) - C2                                               CS1
      DO 30 J=1,L                                                       CS1
      XDP(J)=XDP(J)+XP                                                  CS1
      XYDP(J)=XYDP(J)+XP*Y(I)                                           CS1
   30 XP=XP*XI                                                          CS1
      DO 40 J=LL,LU                                                     CS1
      XDP(J)=XDP(J)+XP                                                  CS1
   40 XP=XP*XI                                                          CS1
C                                                                       CS1
C     STORE ABOVE COMPUTED SUMS IN ARRAY XM                             CS1
C     AND COMPUTE ROW SUMS FOR CHECK SOLUTION                           CS1
C                                                                       CS1
      DO 50 I=1,L                                                       CS1
      LL=I+L                                                            CS1
      YM(LL)=0.0                                                        CS1
      LU=(I-1)*L                                                        CS1
      JK=I-1                                                            CS1
```

```
      DO 50 J=1,L                                                    CS1
      JK=JK+1                                                        CS1
      JC=LU+J                                                        CS1
      XM(JC)=XDP(JK)                                                 CS1
   50 YM(LL)=YM(LL)+XM(JC)                                           CS1
      DO 60 I=1,L                                                    CS1
   60 YM(I)=XYDP(I)                                                  CS1
C                                                                    CS1
C     SOLVE THE SYSTEM XM*C=YM                                       CS1
C                                                                    CS1
      CALL MTXEQ(XM,YM,YM,L,2)                                       CS1
C                                                                    CS1
C     REORDER AND MOVE SOLUTION TO C AND FIND                        CS1
C     MAXIMUM ERROR IN CHECK SOLUTION                                CS1
C                                                                    CS1
      EMEQ=0.0                                                       CS1
      DO 70 I=1,L                                                    CS1
      JK=K-I+2                                                       CS1
      C(JK)=YM(I)                                                    CS1
      JC=I+L                                                         CS1
   70 EMEQ=AMAX1(EMEQ,ABS(YM(JC)-1.0))                               CS1
C                                                                    CS1
C     ADJUST COEFFICIENTS FOR ORIGINAL RANGE                         CS1
C     OF X                                                           CS1
C                                                                    CS1
      DO 80 I=1,K                                                    CS1
      DO 80 J=1,I                                                    CS1
   80 C(J) = C(J) * C1                                               CS1
      C1 = (XMAX + XMIN) / 2.0                                       CS1
      DO 90 I=1,K                                                    CS1
      M=L-I+1                                                        CS1
      DO 90 J=2,M                                                    CS1
   90 C(J) = -C1 * C(J-1) + C(J)                                     CS1
C                                                                    CS1
C     INITIATE PRINT OF ERROR ANALYSIS IF LIST .NE. 0                CS1
C                                                                    CS1
      IF (LIST.EQ.1) PRINT 1001                                      CS1
C                                                                    CS1
C     COMPUTE MAXIMUM AND ROOT MEAN SQUARE ERRORS                    CS1
C     AND OUTPUT ERROR ANALYSIS IF LIST .NE. 0                       CS1
C                                                                    CS1
      EMAX=0.0                                                       CS1
      SUM=0.0                                                        CS1
      DO 130 I=1,N                                                   CS1
      YC=C(1)                                                        CS1
      DO 100 J=1,K                                                   CS1
  100 YC=YC*X(I)+C(J+1)                                              CS1
      DIF=YC-Y(I)                                                    CS1
      IF (LIST.EQ.0) GO TO 120                                       CS1
      IF (I .GT. L)    GO TO 110                                     CS1
      PRINT 1002, I, X(I), Y(I), YC, DIF, C(I)                       CS1
      GO TO 120                                                      CS1
  110 PRINT 1002, I, X(I), Y(I), YC, DIF                             CS1
  120 EMAX=AMAX1(EMAX,ABS(DIF))                                      CS1
  130 SUM=SUM+DIF**2                                                 CS1
```

```
      ERMS=SQRT(SUM/FLOAT(N))                                          CS1
      IF (LIST.EQ.1) PRINT 1003, EMAX, ERMS, EMEQ                      CS1
      RETURN                                                           CS1
C                                                                      CS1
C     GIVE ERROR MESSAGE AND RETURN TO                                 CS1
C     SYSTEM VIA FXEM                                                  CS1
C                                                                      CS1
  200 PRINT 1000, N, K                                                 CS1
      CALL SYSTEM (200,1L )                                            CS1
      RETURN                                                           CS1
 1000 FORMAT (3H0N=,I12,3H K=,I12,29HINCORRECT FOR SUBROUTINE PLSQ)    CS1
 1001 FORMAT (1H1,20X,32HPLSQ     POLYNOMIAL LEAST SQUARE              CS1
     *24HCURVE FIT ERROR ANALYSIS//                                    CS1
     *6H0     I,11X,9HX - GIVEN,11X,9HY - GIVEN,11X,                   CS1
     *10HY - FITTED,12X,5HERROR,16X,4HB(I)//)                          CS1
 1002 FORMAT (1X,I5,8X,5(1PE14.6,6X))                                  CS1
 1003 FORMAT (1H0,9X,5HEMAX=,1PE15.6,9X,5HERMS=,E15.6,                 CS1
     *9X,5HEMEQ=,E15.6)                                                CS1
      END                                                              CS1
```

```
      SUBROUTINE MTXEQ(A,X,B,N,K)                                       CS1
C                                                                       CS1
C     MATRIX EQUATION SOLVER     (7094 FORTRAN IV)                      CS1
C                                                                       CS1
C     USAGE...                                                          CS1
C                                                                       CS1
C     TO SOLVE THE LINEAR SYSTEM     AX=B                               CS1
C                                                                       CS1
C     CALL MTXEQ(A,X,B,N,K)                                             CS1
C                                                                       CS1
C        WHERE A MUST BE DIMENSIONED N X N                              CS1
C              X MUST BE DIMENSIONED N X K                              CS1
C              B MUST BE DIMENSIONED N X K                              CS1
C              N IS THE NO. OF EQUATIONS (ROWS IN A,X,B)                CS1
C              K IS THE NO. OF SOLUTION VECTORS (COLS. IN X,B)          CS1
C                                                                       CS1
C     664 CELLS OF BLANK COMMON ARE USED.                               CS1
C                                                                       CS1
C     NOTE...   TO CHANGE DIMENSIONS OF ARRAYS C AND PIV, ALSO          CS1
C          CHANGE VALUES OF NMAX AND NKMAX IN DATA STATEMENT.           CS1
C                                                                       CS1
C                                                                       CS1
      DIMENSION   A(N,N), B(N,K), X(N,K)                                CS1
      COMMON   ATPE, I, IFROM, IP1, IPIV, ITO,                          CS1
     1         J, KP, L, NP, NP1, NPJ, NPK, RM                          CS1
      COMMON     PIV(26), C(24,26)                                      CS1
      DATA     NMAX, NKMAX/ 24, 26/                                     CS1
C                                                                       CS1
C     GET ARGUMENTS N AND K.                                            CS1
C                                                                       CS1
      NP=N                                                              CS1
      KP=K                                                              CS1
C                                                                       CS1
C       ST N AND K FOR CORRECT RANGE                                    CS1
C                                                                       CS1
      IF(NP.LE.0.OR.NP.GT.NMAX) GO TO 190                               CS1
      IF(KP.LE.0.OR.(NP+KP).GT.NKMAX) GO TO 190                         CS1
C                                                                       CS1
C     MOVE ARRAYS A(I,J) AND B(I,J) INTO C(I,J)                         CS1
C                                                                       CS1
      DO 10 J=1,NP                                                      CS1
      DO 10 I=1,NP                                                      CS1
   10 C(I,J)=A(I,J)                                                     CS1
      DO 20 J=1,KP                                                      CS1
      NPJ=NP+J                                                          CS1
      DO 20 I=1,NP                                                      CS1
   20 C(I,NPJ)=B(I,J)                                                   CS1
C                                                                       CS1
C     SET TO PERFORM N ELIMINATION SWEEPS (I=1,N)                       CS1
C                                                                       CS1
      NP1=NP+1                                                          CS1
      NPK=NP+KP                                                         CS1
      DO 120 I=1,NP                                                     CS1
      IP1=I+1                                                           CS1
C                                                                       CS1
```

```
      C     SEARCH FOR NEXT PIVOT ROW (I-TH PIVOT IS IN COL. I)                CS1
      C                                                                        CS1
            ATPE=0.                                                            CS1
            DO 40 J=I,NP                                                       CS1
            IF (ABS(C(J,I))-ATPE) 40,30,30                                     CS1
       30   ATPE=ABS(C(J,I))                                                   CS1
            IPIV=J                                                             CS1
       40   CONTINUE                                                           CS1
      C                                                                        CS1
      C     OPERATE ON THE PIVOT ROW                                           CS1
      C                                                                        CS1
            IF (ATPE) 210,210,50                                               CS1
       50   DO 60 J=IP1,NPK                                                    CS1
       60   PIV(J)=C(IPIV,J)/C(IPIV,I)                                         CS1
      C                                                                        CS1
      C     PERFORM ELIMINATIONS BELOW THE DIAGONAL (COL. I)                   CS1
      C                                                                        CS1
            IFROM=NP                                                           CS1
            ITO=NP                                                             CS1
       70   IF (IFROM-IPIV) 80,100,80                                          CS1
       80   RM=-C(IFROM,I)                                                     CS1
            DO 90 J=IP1,NPK                                                    CS1
       90   C(ITO,J)=C(IFROM,J)+RM*PIV(J)                                      CS1
            ITO=ITO-1                                                          CS1
      100   IFROM=IFROM-1                                                      CS1
            IF (IFROM-I) 110,70,70                                             CS1
      C                                                                        CS1
      C     PUT THE I-TH PIVOT ROW IN THE VACATED ROW I                        CS1
      C                                                                        CS1
      110   DO 120 J=IP1,NPK                                                   CS1
      120   C(I,J)=PIV(                                                        CS1
      C                                                                        CS1
      C     NOW DO THE BA   SOLUTION                                           CS1
      C                                                                        CS1
            I=NP                                                               CS1
      130   IP1=I                                                              CS1
            I=I-1                                                              CS1
            IF (I) 160,                                                        CS1
      140   DO 150 J=NP1,                                                      CS1
            DO 150 L=IP1,NP                                                    CS1
      150   C(I,J)=C(I,J)-C(I,L)*C(L,J)                                        CS1
            GO TO 130                                                          CS1
      C                                                                        CS1
      C     MOVE THE SOLUTION TO ARRAY X(I,J)                                  CS1
      C                                                                        CS1
      160   DO 170 J=1,KP                                                      CS1
            NPJ=NP+J                                                           CS1
            DO 170 I=1,NP                                                      CS1
      170   X(I,J)=C(I,NPJ)                                                    CS1
      180   RETURN                                                             CS1
      C                                                                        CS1
      190   PRINT 1000, NP, KP                                                 CS1
            CALL SYSTEM (200,1L )                                              CS1
            STOP                                                               CS1
      210   PRINT 1001                                                         CS1
```

```
      CALL SYSTEM (200,1L )                                          CS1
      STOP                                                           CS1
 1000 FORMAT(3H0N=I12,5H    K=I12,35H ARE INCORRECT FOR SUBROUTINE MTXEQ) CS1
 1001 FORMAT (37H0DET(A)=0 IN CALL TO SUBROUTINE MTXEQ)              CS1
      END                                                            CS1
```